

格上げ制限 2 相施錠方式の性能評価

春原 典彦 生山 陽[†] 篠崎 健一^{††} 宮崎 収兄

{sunohara,miyazaki}@mz.cs.it-chiba.ac.jp

千葉工業大学 情報工学科

〒275 千葉県習志野市津田沼 2-17-1

本稿では、データベースにおける並行処理制御について述べる。一般的によく使われている 2 相施錠方式 (2PL) は、条件によってデッドロックが生じ性能が低下する問題がある。保守的スケジューラを導入した 2 相施錠方式や、楽観的なスケジューラなどの他の方式も検討されているがあまり使われていない。我々は一部のデッドロックしか防止できないが保守的 2 相施錠より柔軟な格上げ制限 2 相施錠方式 (NU2PL) を提案した。本稿では格上げ制限 2 相施錠方式の性能について検討を行い、2 相施錠方式で発生する大部分のデッドロックを本方式により防止できることを示す。

Performance of Non-Upgrading Two-Phase Locking

Norihiko Sunohara, You Ikuyama[†],
Kenichi Shinozaki^{††}, Nobuyoshi Miyazaki

Department of Computer Science, Chiba Institute of Technology
2-17-1 Tsudanuma Narashino Chiba 275 Japan

This paper discusses concurrency control mechanisms for databases. The most widely used method, the two-phase locking, may have performance problem due to the deadlock under certain conditions. Other methods including the cautious two-phase locking and optimistic schedulers are discussed more adequate for some applications, but are not widely used. We proposed the non-upgrading two-phase locking which is more flexible than the cautious two-phase locking although it can prevent only certain types of deadlocks. This paper discusses performance evaluation of non-upgrading two-phase locking and shows that it is able to effectively prevent most of the deadlocks in two-phase locking.

[†] : 現在 日立ソフトウェアエンジニアリング株式会社

^{††} : 現在 株式会社メイテック

1 はじめに

データベースシステムには異なるトランザクションからの同時的なアクセスに対し、その実行を制御するための並行処理制御(concurrency control)が必要不可欠である。この制御方式にはいくつかの方法が提案されていて、施錠をかける方法やスケジューラで管理する方法などがある。その中でも現在は使いやすさの面から2相施錠方式(two-phase locking)が使われている。しかし、2相施錠方式においても高負荷状態ではトランザクション完了率が低くなってしまいうため、問題が生じる場合がある。この問題には保守的スケジューラの導入が有効であると言われている[NTI89]。しかし2相施錠方式の保守的スケジューラ(C2PL)では、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すべて施錠する必要があり、実際の応用に適用するのは困難な場合が多い。また、楽観的スケジューラ(CPS)も提案されている[BHG89]が、負荷が高い場合は完了率が低くなるといった問題がある。そこで我々は2相施錠方式に注目し改良を行うことで新たな施錠方式を提案した。それが格上げ制限2相施錠方式(Non-Upgrading Two-Phase Locking)である[春原95]。

本稿では、我々の提案した格上げ制限2相施錠方式(以下NU2PLとする)と原型となる2相施錠方式(以下2PLとする)とを比較・検討するために簡単なシミュレーターを作成し、NU2PLが完了率・並行性について2PLよりも優れていることを示す。2章ではNU2PLについて簡単な説明を、3章では簡単なモデルでの解析的評価、4章ではシミュレーターを使った一般的な条件のもとでの評価をし、5章で比較・検討を行う。

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りやすい。格上げ制限2相施錠方式(NU2PL)ではこのようなデッドロックを防止するために以下の制限を取り入れた。

[格上げ制限規則]

・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

言い換えると後で、writeする場合は、最初のreadの時に排他施錠を行うようにする。

2.2 NU2PLの動作

以下にreadした後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。

・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が考えられる。[谷口94]などによれば高負荷状態では2PLとCPS（競合保存直列可能スケジューラー）は完了率が低くなるに従ってスループットも低下するので、NU2PLではスループットの向上が期待できる。

2.3 NU2PLにより防止できるデッドロック

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

<2PL>

$$D(2PL) = (R_i \cap W_j \neq \phi) \wedge (R_j \cap W_i \neq \phi)$$

<NU2PL>

D(NU2PL)=

$$\begin{aligned} D(2PL) \wedge ((R_i \cap R_j \cap W_j) = \phi) \wedge ((R_j \cap R_i \cap W_i) = \phi) \\ = ((R_i \cap W_j) \cap R_j^c \neq \phi) \wedge \\ ((R_j \cap W_i) \cap R_i^c \neq \phi) \end{aligned}$$

上記の式から以下のようなデッドロックを防げることがわかる。

- ・ write対象がread対象に含まれる場合
($W_i \subset R_j$)

- ・ 2つのトランザクションが同一の物を read, write する場合 ($R_i = R_j = W_i = W_j$)

先ほど述べたようにread対象とwrite対象は何らかの関係を持っていることが多いので、このようなデッドロックを防ぐことが有効と考えられる。

3 デッドロック防止率の解析的評価

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ために以下のような単純なモデルを考える。

[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 $1/n$ とする。
- ・ 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。

このモデルをもとに、2PLでデッドロックが起こる状態を考えると次の2通りであることがわかる。

$$\begin{aligned} \text{タイプ (1)} \quad T_1 : r [i] \rightarrow w [i] \\ T_2 : r [i] \rightarrow w [i] \end{aligned}$$

$$\begin{aligned} \text{タイプ (2)} \quad T_1 : r [i] \rightarrow w [j] \\ T_2 : r [j] \rightarrow w [i] \end{aligned}$$

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を目的にしているためデッドロック防止率は次の式で表す事ができる。

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図1のようになる。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。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の大部分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。

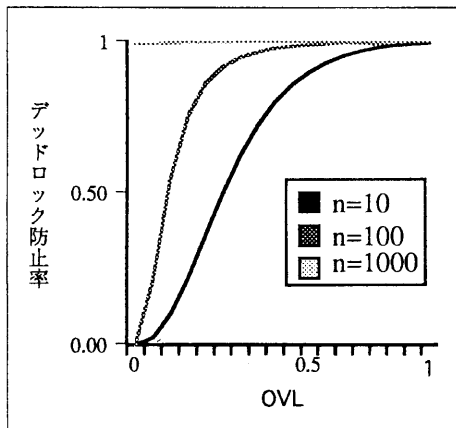


図1 定量評価におけるデッドロック防止率

4 シミュレーションによる評価

4.1 シミュレーションのモデル

readした後writeするという2段階トランザクションはそのままに、より一般的な評価をシミュ

レーションで行う。解析的評価ではreadまたはwriteするデータ数を1つに限定したが、アクセスするデータ数すなわち、read施錠数・write施錠数をパラメータとして設定できるようにした。

4.2 シミュレーションのパラメーター

発生間隔、readサービス時間、writeサービス時間は指数分布とする。また、アクセスするデータ項目は一様分布とする。主なシステムパラメーターを以下のように定義する。

(1) OVL (オーバーラップ率) : read対象とwrite対象が一致する確率を示し0~1の値をとる。

(2) n (データ項目数) : 実験対象のデータ項目数。

(3) IT (発生間隔) : トランザクションの平均発生間隔。この時間がRT・WTに対し短いほど制御に対し負荷がかかることとなる。

(4) RT (readサービス時間) : トランザクションがデータをreadするのにかかる平均時間。

(5) WT (writeサービス時間) : トランザクションがデータをwriteするのにかかる平均時間。

(6) RL (read施錠数) : 1トランザクションあたりのread施錠をする項目数。

(7) WL (writeの施錠数) : 1トランザクションあたりのwrite施錠をする項目数。

また、TR (実験トランザクション数) は10万回とする。

4.3 評価

以下のパラメータにおける2PLとNU2PLのデッドロック数の推移を図2に示す。

[パラメータ]

$$n=1000, IT=3, RT=3, WT=3, RL=3, WL=3$$

このグラフをみてもわかるように、2PLではOVLが増加するとデッドロック数は確実に増えていることがわかる。OVL=0とOVL=1とを比較すると、

デッドロック数は約8倍にもなる。しかし、それとは対照的にNU2PLではOVLの増加と共にデッドロックが減少している。これはread対象とwrite対象が一致する場合のデッドロックを防いでいるNU2PLの効果の現れである。OVL=0の場合はNU2PLは2PLと同様な、OVL=1の場合はC2PLと同様な特性をもつことがわかる。

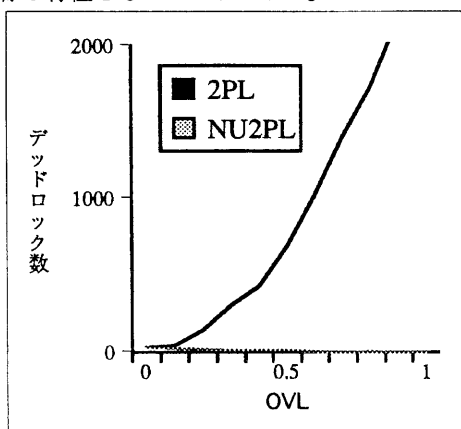


図2 n=1000のデッドロック数の推移

次に前節の解析的評価をしたものと比べるために、データ数を10,100,1000としRL=WL=1としてデッドロック数を検出し、デッドロック防止率について図3に示す。この図を見てもわかるように、解析的評価のグラフとほぼ一致し、NU2PLが2PLより優れている。

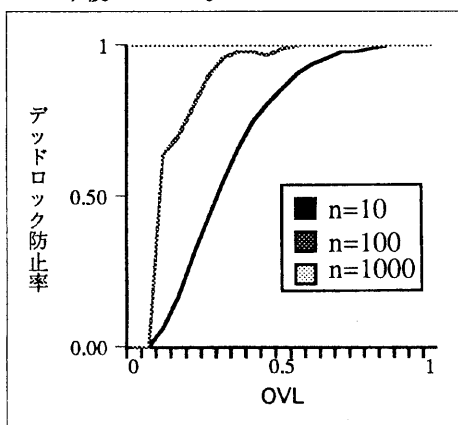


図3 シミュレータにおけるデッドロック防止率

次に、負荷によるデッドロック数の違いについて3つのCASEでシミュレーションを行う。ここでいう負荷とはRTとWTに対するITの関係すなわち、サービス時間とトランザクションの発生間隔の比からなるものであり、以下にそれぞれのケースについて説明する。まずCASE1を基準としてパラメータを[n=5000, IT=3, RT=3, WT=3, RL=3, WL=3]とする。それに対しCASE2、CASE3ではITの値を変えることで負荷を変える。まずCASE2ではIT=1とする。これはRT,WTに対してITの値を小さくし、負荷をCASE1に比べ多くかけた。CASE3ではIT=6とし、CASE1よりも負荷を小さくした。それぞれのCASEのデッドロック数の推移(2PL)を図4に示す。見てもわかるように2PLでは負荷によりデッドロック数に大きな差が現れる。しかしNU2PLではグラフには示していないがデッドロック数はほぼ0となりほとんどのデッドロックを防いでいることがわかる。つまり高負荷状態においてNU2PLが優れていることを示せた。

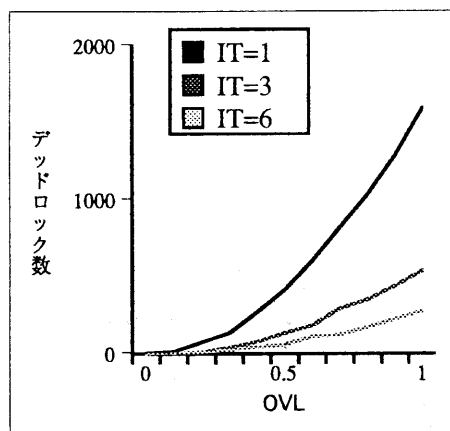


図4 負荷とデッドロック数 (2PL)

4.4 多粒度施錠の評価

NU2PLは多粒度施錠へも拡張可能である[春原95]。多粒度施錠におけるNU2PLの効果を検討す

る。階層化されたデータに対しての施錠を考えるため多数のデータに対してread施錠をかけ、その中の少数に対しwrite施錠をかけることで擬似的な評価を行うことができる。つまり以下のようなパラメータでシミュレーションを行う。

[パラメータ]

n=1000, IT=3, RT=3, WT=3, RL=10, WL=1

結果を図5に示す。擬似的な評価であるが、図を見てもわかるようにOVL=0.5を超える範囲ではNU2PLのデッドロック数はほぼ0になり2PLで発生していたデッドロックを防止できていることがわかる。つまり多粒度施錠に対してもNU2PLが優れた性能を示すことができた。また、多粒度NU2PLでクラスをインスタンスの上位と考えることにより、オブジェクト指向データベースにおけるクラストランザクションとインスタストランザクションの競合を防止するために提案された遅延式CPS[谷口93]と同様な効果を得ることが期待できる。

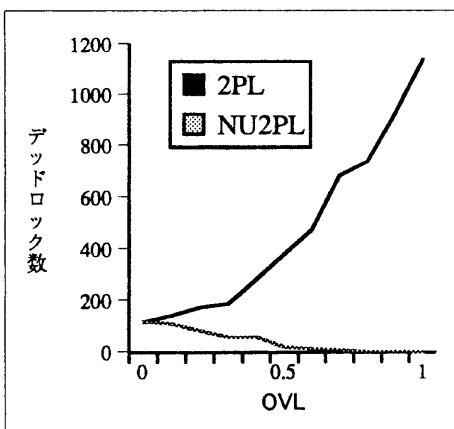


図5 多粒度施錠における
デッドロック数の推移

5 検討

簡単なモデルのシミュレーションにより、我々の提案したNU2PLの性能が2PLよりも優れている

ことを示せた。特に高負荷状態においても十分な性能を発揮できた。このことで高負荷状態におけるトランザクション完了率の問題も解決できるはずである。またNU2PLは2PLの改良版であるため、デッドロックが頻繁に起きるトランザクションのみNU2PLを、2PLと併用することが可能であるため柔軟性も優れている。

<参考文献>

- [NTI89] Nishio, S., Taniguchi, S., and Ibaraki, T., On the Efficiency of Cautious Schedulers for Database Concurrency Control - Why Insist on Two-Phase Locking?, J. Real-Time Systems, Vol.1, pp177-195, Kluwer Academic Publishers, 1989
- [BHG89] Bernstein P.A., Hadzilacos V. and Goodman N.: "Concurrency Control and Recovery in Database Systems", Addison-Wesley(1989)
- [谷口94] 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 (1994)
- [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)