

データ蓄積処理における並列性の向上

— 実現方式 —

3R-5

深沢 信人, 浦 満広, 伊達 政広
(富士通株式会社)

1. はじめに

本稿では, OLTP (Online Transaction Processing) 処理におけるデータ蓄積処理の並列性を向上させるための実現方法を報告する。

従来のデータベースシステムでは, 図1に示すようにデータの発生した時よりトランザクションが終了するまでの間ファイルのロックを獲得することによって, データ間の不整合が発生しないようにしていたために, データ蓄積処理の並列性が犠牲となっていた。しかし, 現在のデータ蓄積処理への要件からすると, 従来とは逆にデータ間の整合性を多少犠牲にしてもデータ蓄積処理の並列性を向上させる必要が生じている [1]。

現状では, データ蓄積処理の並列性の向上はアプリケーションによって個々に実現されているが, 商用DBMSで実現するための具体的な実現方式を説明する。

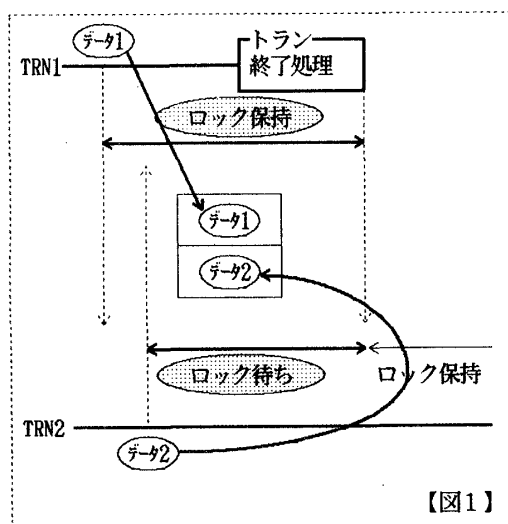
2. 並列性向上の手法

ここでは, データを格納する際に獲得するファイルのロックの保持期間を短くすることによって, データ蓄積処理の並列性を向上させる方式について述べる。

この方式は, トランザクション中で発生したデータをトランザクションごとに用意したバッファに一時的に蓄積しておき, あるタイミングでバッファ内のデータを一括してファイルに格納する方式である。以後この方式を Buffered Heap 方式と呼ぶ。

Buffered Heap 方式は, ファイルのロック期間を短くしてデータ蓄積処理の並列性を向上させ, またバッファ内のデータをファイルに格納するタイミングを選択できるという特長がある。

以降, この方式の特長に着目して Buffered Heap方式を説明する。



【図1】

方式1) トランザクション終了処理中にファイルへ格納

トランザクション終了処理のデータ格納処理で, ファイルのロックを獲得し, バッファに蓄積していたデータをファイルに格納してロックを解放する。(図2)

方式2) トランザクション終了後にファイルへ格納

トランザクションの終了処理にてデータを蓄積したバッファをトランザクション終了後に持ち越し, トランザクション終了処理をデータの格納処理をしないで終わらせる。バッファは, トランザクションとは別のタイミングでファイルに格納する。(図3)

3. データ蓄積処理要件との対応

方式1の場合、トランザクションが終了したデータを必ずファイルに格納するので、ファイルを入力するアプリケーションには最新データがかならず参照可能である。

方式2の場合、トランザクションが終了してもバッファ上にデータが残っている可能性があるため、ファイルを入力するアプリケーションが必ず最新データを参照できるという保証はない。

表1に並列性向上のための要件との対応を示す。

【表1】

並列性向上のための要件分析	方式1	方式2
トランザクション終了後のすべてのデータを参照	○	×
トランザクション終了後のデータの遅延参照	○	○
トランザクション終了前のデータを参照	×	×
トランザクション終了順にデータを参照	○	△
トランザクション終了順と多少前後してデータを参照	○	○
トランザクション終了順と無関係にデータを参照	○	○

- ：要件を満たす
- ×：要件を満たさない
- △：実現方法によっては要件を満たす

4. おわりに

本稿では、データ蓄積処理の並列性を上げるための実現例を報告し、それらがデータ参照処理の要件の分析によって裏付けられていることを示した。今後の商用DBMSでは、このことをふまえて、データ参照処理の要件に応じたインテグリティレベルの実現を行っていく。

今後は、データ蓄積処理の並列性の向上をみるためのデータを収集し、具体的な効果をはかっていきたい。

<参考文献>

- 〔1〕 浦満広, 深沢信人, 弘末清悟: データ蓄積処理における並列性の向上—問題分析—, 本紙

