

トランザクションマシンにおける  
ハードウェアによる並行処理

## 7Q-2

上林彌彦 中村千秋 最所圭三  
九州大学 工学部**1. はじめに**

本研究ではデータベースを高速にアクセスするマシンであるトランザクションマシンを考え、このマシンを効率よく動かすためのハードウェアによる並行処理制御について検討を行なう。

情報化社会の発達にともないデータベースが扱うデータ量は著しく増大してきている。この大量のデータを高速に扱うための専用マシンとしてデータベースマシンがある。これまでデータベースマシンについて様々な研究がなされてきているが、多くの研究では整列器やフィルタ等をハードウェアを用いて高速化する手法が主である。これに対して本稿で述べるトランザクションマシンではデータアクセスの高速化およびシステムの高信頼性に特徴がある。本研究では、制御のオーバーヘッドを減らすためにハードウェアを用いる並行処理制御の設計を行っている。

**2. 基本的事項****2. 1 トランザクションマシン**

商用に使われているトランザクションシステムとして、バンキングシステムなどがある。このようなシステムは

- ・一度に多くのアクセス要求を受ける。
- ・各アクセス要求が同じデータにぶつかることは少ない。
- ・データの更新が主であり、結合、整列などの演算が行われることは少ない。

などの特徴を持つ。このようなシステムを構成するトランザクションマシンでは、従来のデータベースマシンのような結合、整列などの演算の高速化よりも、データアクセスの高速化、信頼性の向上が中心課題となる。そのため本研究では、入力されるトランザクションの基本操作である読み込み、書込み、確定の高速化に特化したトランザクションマシンを考える。

**2. 2 並行処理制御方式**

並行処理とは、複数のトランザクションをシャッフルしながら実行することによってCPUとI/Oを効率よく使うための手法である。この時、実行されたスケジュールが直列可能性(処理結果が正当である)を満足しなければならない。直列可能性を保証する代表的な方式として2相施錠方式がある。

**2相施錠方式<sup>[1]</sup>**

あるトランザクション $T_i$ がデータ $X$ をアクセスしようとするとき、そのデータを施錠することによって、他のトランザクションからのアクセスを禁止してしまえる。2相施錠方式では、直列可能性を満足させるため、施錠段階と解錠段階が完全に別れているものである。この方式は施錠を用いるために、トランザクションが互いに必要とするデータを施錠することにより、それ以上処理を進められなくなる状態になることがある。これを

すくみと呼ぶ。このすくみを解消するためには、すくみに入っているトランザクションを1つ以上取りやめなければならない。

**3. 並行処理制御の問題点とハードウェア化の必要性**

以下の2つの理由により、並行処理制御によるオーバーヘッドがシステムのスループットに大きく影響する。

まず第1点として、データアクセスの高速化である。従来のディスクベースのシステムでは、ディスクからデータを持ってくる時間と直列可能性の判定を行う時間を比較した場合、最近のプロセッサの処理能力から考えると、割合としてディスクアクセスの時間が大きい。そのため、直列可能性の判定に費やされる時間はディスクアクセスの時間に隠されてしまう。これを、データアクセスを速くするために大容量主記憶を使用することを考えると、ディスクベースシステムの時は逆に直列可能性判定の時間にメモリアccessの時間が隠されてしまう。そのためシステムのスループットに並行処理制御のオーバーヘッドが大きく効いてくる。

第2点として、従来のディスクシステムの場合のデータアクセス単位は、ディスクのトラック単位のように大きかった。これは、ディスクがトラック単位でのアクセスが速いためである。しかし、このような大きい単位ではトランザクションのデータアクセスの競合が起き易い。そのため並行度が下がってしまう。しかし、主記憶を用いたデータベースの場合、メモリ上のどのデータに対しても同じ時間でアクセスすることができ、アクセス時間はデータ量に比例する。このため、データアクセス単位をもっと小さくすることができる。これによって並行度があがる。しかし、排他制御の単位を小さくしたために、すくみの検出に多くの時間を使用してしまう。このため、並行処理制御のオーバーヘッドが大きくなり、スループットを低くしてしまう。

以上のような問題点の解決にはどうしてもソフトウェアによって解決することは難しい。本稿では、この問題点の解決のためにハードウェアによって並行処理を行うことでオーバーヘッドを小さくすることを目指す。

**4. トランザクションマシンの構成**

図1のようなトランザクションマシンのもとでハードウェアによる並行処理制御を考える。このトランザクションマシンでは処理の簡単化のために、すくみの検出にタイムアウトを用いた2相施錠方式を用いる。

**4. 1 トランザクション管理部 TM:**

トランザクションマシンと外部とのインタフェースをとる部分である。入力として、読み込み、書込み、確定からなるトランザクションを一括して受け取る。出力としては、入力したトランザクションの中の操作の順にデータを逐次返していく。また、そのトランザクションが後退復帰した

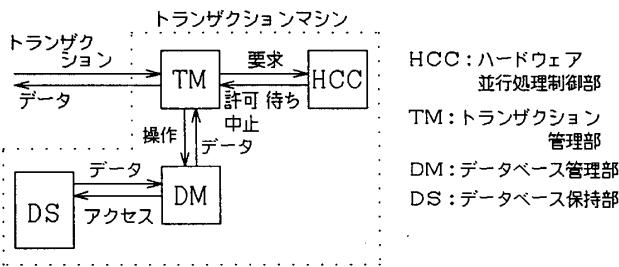


図1 トランザクションマシン

ことも出力する。内部では入力されたトランザクションのアクセス要求をハードウェア並行処理制御部(HCC)に送り、その応答によってデータのアクセス操作をデータベース管理部(DM)へ送ったりトランザクションの後退復帰を行ったりする。

4. 2 データベース管理部 DM:

TMがHCCからアクセス許可をもらった後、送ってくるデータアクセス操作を行う部分。また、論理的なアクセス操作を物理的なアクセス操作に変換を行う。

4. 3 データベース保持部 DS:

データベースを蓄えておくところ。ここでは大容量主記憶を使用している。

4. 4 ハードウェア並行処理制御部 HCC:

入力はTMからのアクセス要求を逐次受け取る。出力は'許可'、'待ち'、'中止'である。このHCCはタイムアウトを用いた2相施錠方式を用いて並行処理制御を行う。また、制御部を高速化するために複数個の施錠検査器を用いて並列に処理を行う。この時、HCCのデータベースの管理には、データベースを分割して、それぞれを施錠検査器に割り当てて管理する方法と、データベースの施錠情報を共有情報として複数の施錠検査器に共有させて、それぞれ検査器にデータベース全体を管理させる方法がある。後者の場合、共有情報を持つために排他制御を行わなくてはならないために通信など、その他のオーバーヘッドが大きくなる。前者の場合、各施錠検査器が施錠検査を行うときに干渉しあうことはない。そのためここでは前者の方法を用いる。

4. 5 トランザクションマシンの動作

TMは外部からトランザクションを受け、そのアクセス要求をHCCに送る。HCCはTMから受け取ったアクセス要求が施錠できるかどうかを判定する。その間TMはDMへのアクセスを行っておく。読み込みに関してはDMを通してデータをTMに保持しておく。書き込みに関してはDMの作業領域に書き込んでおく。書き込みはそのトランザクションが確定されないと、実際にDSへの書き込みは行わない。HCCから'許可'が送られてきた場合、TMは読んでおいたデータを外部へ出力する。また書き込みに関しては、確定操作が行われるまでそのままにしておく。HCCから'待ち'が帰ってきた場合、読んでおいたデータは捨てる。また、書き込んだデータも無効にする。HCCから中止が帰ってきた場合、読み込んでおいたデータも書き込んでおいたデータも捨て、さらに外部に後退復帰がおこったことを出力し、そのトランザクションをやり直す。

5. ハードウェア並行処理制御部の内部

HCCは図2のように、施錠検査を行う施錠検査器(LC)、TMとのインタフェースをとるトランザクションインタフェース(TI)、施錠ができなかった場合のタイマの設定、タイムアウトの検出、タイムアウトによってTMへ、トランザクシ

ョンの'中止'を送るタイムアウト検出器(TC)からなっている。

本研究においてはHCCの各部をハードウェアによって構成することを考える。これによってLCとTCの処理を並行して処理することができるが、解錠、後退復帰などを行う場合には通信が必要となる。

HCCの内部動作を見ると、アクセス要求が入ってくると、TIが適当なLCにアクセス要求を渡す。渡されたLCは施錠可能かどうか検査する。施錠できればLCの持つローカルメモリに施錠情報を書き込み、TMへ'許可'を送る。施錠可能でなければTMへ'待ち'を送り、そのアクセス要求をTCへ渡す。受け取ったTCは時刻印をつけ、ローカルメモリ内の待ち行列にに入れる。TCは現在の時刻と時刻印を比較しながらタイムアウトを検出する。また、トランザクションが確定するか、後退復帰したために解錠されたデータの施錠を待っているアクセス要求がないか解錠が起きる度に待ち行列を調べる。もしあれば待ち行列から抜き出し、再度LCへ入力される。TCがタイムアウトを検出した場合、そのトランザクションを後退復帰させるためにTMへ'中止'を送り、待ち行列からそのトランザクションのアクセス要求を取り除くと共に、そのトランザクションが施錠しているデータをLCに解錠させる。

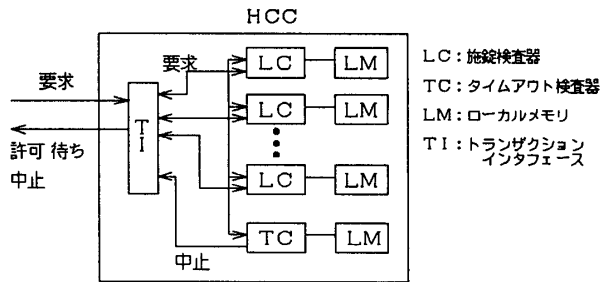


図2 HCCの内部

4. まとめ

本稿では、トランザクションマシンについて説明を行い、ハードウェアによる並行処理制御として2相施錠方式を使用する方法について述べた。今回は2相施錠を用いて述べたが、データのアクセス単位を小さくすることで競合が少なくなる場合は、事後検証方式の効率を向上できる可能性を持つ。問題となるのはHCCの内部の並列処理の効率である。

今後の研究ではトランザクションマシンを試作し、まず第一段階として、CPUボードを用いて並行処理制御部をシミュレーションする。それによって各方式の検討を行う。また、大容量主記憶を用いることから何らかのバックアップが必要となるが、この部分の開発も行っていく予定である。

謝辞

本研究にあたり、有益な御助言をいただいた本学 古川哲也助手、ならびに掛下哲郎氏をはじめとする研究室諸氏に深謝いたします。

尚、本研究の一部は文部省科研費によるものである。

参考文献

[1] Bernstein, P.A., Hadzilacos, V., Goodman, N., Addison-Wesley, Concurrency Control and Recovery in Database Systems, 1987