1G-6

関係データベースの更新処理に於けるロック方式の一考察

福田晃，吉田浩，喜連川優，田中英彦
元岡達（東京大学　工学部）

1　はじめに

本論では、ロックのレベルをリレーションよりも小さくして、属性，タップルレベルにした時にどのような問題が生じるかを考えてみる。タップルレベルにした時に、ファントム[5]の存在を考慮しなければならないことは、K.P. Eswaranらによって、示されている。そして、その解決のためにプレディケイトロックが提案されている。ここでは、属性レベルの考察とプレディケイトロックの可能性を考察する。

2　プレディケイトロックの実際

まず、トランザクションT_iは、検索，更新，追加，削除と分類して、単純なものを考える。また、ロックとは、

「２つ以上のトランザクションが同時実行されることによって生じる意味上の矛盾を、事前に防ぐこと。」

とする。また、各トランザクションは、正当なもので、逐次実行すれば矛盾は生じないものとする。

ここで、普通の問い合わせを考えてみると、次のようになる。追加以外の問い合わせは、何らかの条件を指定して、それに該当するタップルのある値を読み出すか、書き替えるか、削除するかに分けられる。

図１．　２つのトランザクションと属性

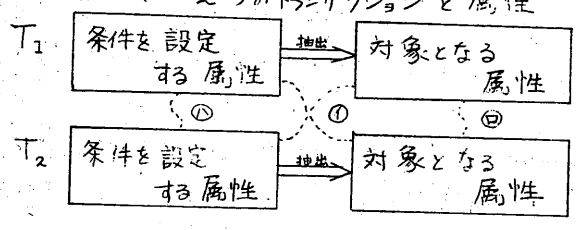

表１　リレーションの例

A	B	C	D	E
A1	100	C1	2	10
A2	200	C2	4	20
A3	150	C2	3	10
A4	170	C1	1	10
A5	130	C3	2	40
A6	180	C1	7	30

条件は、属性によって指定するので、条件に要求される属性と、操作の対象となる属性を考えてみる。各属性は、関数従属などによる相互の制約がない限り独立であるから、別々に操作されてもよい。この状況は、図１のようになる。例えば、表１の関係に対し次の２つのトランザクションを考える。

T_1；A=A1 であるタップルのBの値を２倍する

T_2；D=2 であるタップルのCの値をC2にする

このT_1, T_2は、属性B,C間に制約がなければ、同時に実行してもよい。問題となるのは、図１の㊁のように、一方の対象となる属性が他方のそれとなる場合と、㋑のように、一方の条件となる属性が他方の処理対象となって操作される場合である。㋑の場合にファントムが問題になる。

・対象に重なりがある場合は、次のようになる。（㊁）

T_1；B＞100 であるタップルのDの値を２倍する。

T_2；E＞20 であるタップルのDの値を１だけ増加させる。

この場合は、B＞100 かつ E＞20 であるタップルが実在しなければよい。（オブジェクトロック可）

・条件と対象が重なっている場合には、データ操作によって、他のトランザクショ

以下にその説明をする。なお，PUはプロセッサユニット，MUはメモリユニットである。

- CSMA：通信状態管理領域．CCPの状態の遷移に従いCCPにより書込まれ，ホストはこの表により通信の状態を知る。
- CNB：コマンド掲示板．CCPへのコマンドはここへ書込まれ，CCPが読出す。
- CRT：通信登録表．通信相手プロセスのサイト，プロセス番等の情報を格納する。
- CMT：通信管理表．通信要求の有無，メッセージバッファのアドレス等を格納する。

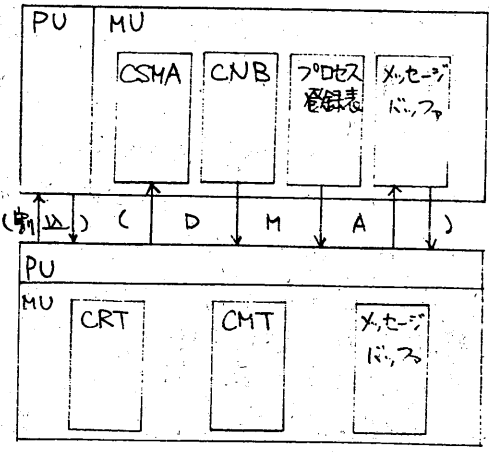

図 2

図に示す様に，データの転送は，ホストCPUに対するDMAにより行なう。これはホストCPUを煩らわせないことと，高速性を考慮したものである。

5．システム構成についての評価及び問題点

以上の様なシステム構成をとったことによる意義及び今後の検討が必要な問題点について，以下に列挙する。

①システム構成上の階層構造を，マルチマイクロプロセッサという形でハードウェア構成上の階層化に投影したことにより，システム把握が容易になった。

②ホスト計算機は，単純なメッセージのみによってCCPを起動可能であり，ホストの通信に関する負荷が著しく軽減された。これは，計算機アーキテクチャ上，大きな意味を持っている。又，機能を分散することにより，個々のレベルでの必要な手順は増加しているが，総括的に考えた場合，並列化によるシステム全体のスループットの向上を図ることができる。

③各ユニット間のインタフェースを標準化したので，従来OSの一部として記述されていたプロセス間通信機構をCCPハードウェアとして分離することにより，計算機ホストにこのCCPを接続することでプロセス間通信機構が容易に実現される。

④試作ではCCPにMOSタイプのマイクロプロセッサZ80Aを用いたが，これのソフトウェアにより，対ホストインタフェースを実現するを，1 Byteの転送につき数10 μsecを要し，高速とはいえない。バイポーラタイプのビットスライス型プロセッサを用いるか，或いは純粋なハードウェア化をすることで，高速にする必要がある。

⑤プロセスに複数の通信ポートを設けることは有用である。

6．おわりに

以上，プロセス間通信機構をOSから分離し，ハードウェア化する方法について検討したわけであるが，システム全てが完全な形では実装されていないので，細かい点について評価，検討が行なわれていない。ハードウェアの試作も，一例であって，さらによい方法の検討も必要である。なおこのシステム構成は，現在作成中のOS記述言語N－Pascal[*]にインタフェースを極力合わせることを考えた。

参考文献

*堀 健一 他 "分散処理向き高級言語 N－Pascal の検討" 本予稿集 5K-6

和賀井フミ子，田中英彦，元岡達，"網向きオペレーティングシステムについての一考察"，EC77-43