データベースマシンＧＲＡＣＥの同時実行制御

5H-5

伏見信也[1]，西村健[2]，喜連川優[2]，田中英彦[2]
[1]三菱電機　[2]東京大学

1．はじめに

並列関係データベースマシンＧＲＡＣＥに於ける同時実行制御について，そのアルゴリズム，実装方式，性能評価の概要について述べる。ＧＲＡＣＥでは，データストリームモデルと呼ばれる計算モデルに基づき，関係演算処理をタスクと呼ばれる統一的な計算構造によって実行する。ＧＲＡＣＥに於ける更新演算も，検索演算と完全に対称的な形でこのタスク構造により実行される。また，その同時実行制御は更新タプルのストリームに対するフィルタリングにより効率良く実現できる［１］。

2．同時実行制御アルゴリズム

ＧＲＡＣＥに於ける同時実行制御の設計に際しては，(1)更新演算，更にはその同時実行制御を複数台のディスクモジュールにより並列実行できること，(2)データストリーム処理に適したアルゴリズムであること，等を目標とした。

ＧＲＡＣＥに於ける同時実行制御アルゴリズムは，２相ロックの変形であるprecision lock［２］に基づく。この方法では，readアクセスに対しては，アクセスに用いられる検索述語を用いた述語ロックを行い，write アクセスに関しては物理ロックを用いる。ここで特にwrite アクセスに関しては，実際にディスク上のデータを変更する前に一度当該データを読み出すことを仮定する。これは実際のデータベース処理に於ては自然な仮定と考えられる。これにより，write アクセス間のコンフリクトチェックは不要となる。

precision lockは，(1)lock granularityが可変である，phantom freeである，ロック要求の比較回数が減少しコンフリクトチェック時間が短縮される，等の述語ロックの利点を保持すると共に，(2)述語ロックのみ用いた場合に生じる一階述語論理の非決定性からの問題がなく，(3)後述の様にＧＲＡＣＥに於けるデータストリーム処理，タスク構造との親和性が極めて高い，等の多くの利点を有する。

3．実装方式

同時実行制御は，大きくロック要求間のコンフリクトチェックと，コンフリクトしたトランザクション間でのデッドロック制御に分れる。この内，デッドロック制御は分散化が困難であり，ＧＲＡＣＥの制御モジュールに集中化されるが，コンフリクトチェックはデータベースを格納しているディスクモジュール群に分散される。

3.1 ディスクモジュールの構成

ディスクモジュールの構成を図１に示す。ディスクモジュールは，データベースを格納するディスク，及びディスクキャッシュと，２つのフィルタプロセッサ，ハッシングユニット，多次元クラスタリングプロセッサ，４つのバッファ，更に障害回復用のログバッファ，ログディスク等からなる。４つのバッファの内，ＰＢ (Predicate Buffer) 及びＵＴＢ(Updated Tuple Buffer)には各々実行されたがコミットされていないトランザクションが発行した検索述語，更新タプル群が格納される。ＷＰＢ(Waiting Predicate Buffer)及びＷＴＢ(Waiting Tuple Buffer)には，コンフリクトの為，実行を待たされている述語群，タプル群が各々格納される。

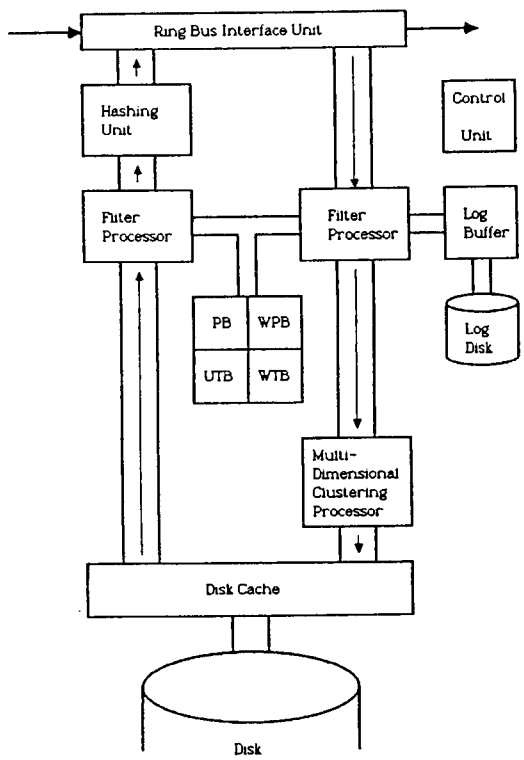

Figure.1　Organization of Disk Module

Concurrency Control in Database Machine GRACE
Shinya Fushimi[1], Ken Nishimura[2], Masaru Kitsuregawa[2], and Hidehiko Tanaka[2]
[1]MITSUBISHI Electric CO., [2]Univ. of Tokyo

3.2 コンフリクトチェック

データベースに対する検索要求は，基本的にディスク（ディスクキャッシュ）→フィルタプロセッサ→ハッシングユニット→リングバスインタフェースユニットとデータストリームを流すことにより実行される。フィルタプロセッサには実行に先立って検索述語がセットされる。この際，ＵＴＢ内のコミットされていないトランザクションが生成した更新タプルがフィルタに通され，この述語が他の更新要求とコンフリクトしていないことが確認される。アクセスコンフリクトの発生は，フィルタ内の述語をＵＴＢ内のタプルが満たすことにより検出できる。コンフリクトが発見されると，当該述語はＷＰＢ内に格納され，コンフリクトしているトランザクションの組が制御モジュールに報告される。コンフリクトが存在しなければ，この述語の使用が許可され，ディスク（ディスクキャッシュ）に対してデータストリームの送出が命令される。フィルタプロセッサは送られてくるデータストリームの中から検索述語を満たすタプルのみを取り出す。ハッシングユニットは，検索結果のタプル群に対してハッシュを施し，次演算に関して動的クラスタリングを行う。

データベースに対する更新要求は，検索要求と完全に対称的に実行される。即ち，更新されたタプルはストリームとなってリングバスインタフェースユニット→フィルタプロセッサ→多次元クラスタリングプロセッサ→ディスク（ディスクキャッシュ）の順にディスクモジュール内を流れる。ここでフィルタプロセッサにはＰＢ内の述語がセットされ，送られてくる更新タプル(exclusive lock)が他のトランザクションの検索要求(read lock) とコンフリクトしていないことが確認される。コンフリクトが検出されると，更新タプルはＷＴＢ内に格納され，検索の場合と同様にコンフリクトが制御モジュールに報告される。コンフリクトが検出されなければタプルは多次元クラスタリングプロセッサに送られ，静的クラスタリングを施されてディスク（ディスクキャッシュ）に格納される。

3.3 デッドロック管理

制御モジュールは，wait-for-graphを管理し，ディスクモジュールから報告されてくるアクセス要求のコンフリクト間のデッドロックを監視する。デッドロックの検出は，デッドロックしているトランザクションの一つから始めて，wait-for-graph上の親を辿ることにより容易に実現できる。デッドロックが検出されると，制御モジュールは新しい方のトランザクションのアボートをディスクモジュールに指示する。

4．タスク構造

ＧＲＡＣＥでは，データベースに対する検索要求はreadタスクとして，更新要求はwrite タスクとして実行される。一般に，ＧＲＡＣＥに於けるタスクは，データストリームの流出空間，フィルタリング処理，クラスタリング処理，データストリームの流入空間から構成される。上記の実装方式では，このタスク構造は，

［readタスク］

流出空間	←→	ディスク群
フィルタリング	←→	フィルタプロセッサ群
クラスタリング	←→	ハッシングユニット群
流入空間	←→	メモリモジュール群

［write タスク］

流出空間	←→	メモリモジュール群
フィルタリング	←→	フィルタプロセッサ群
クラスタリング	←→	多次元クラスタリングプロセッサ群
流入空間	←→	ディスク群

によって実現されている。この様に，各演算を共通のタスク構造により実行することにより，システムソフトウェアは，統一的な資源管理，実行管理等を行うことができる。

5．性能評価［３］

以上の同時実行制御アルゴリズムに関し，シミュレーションにより性能評価を行った。ここで，マシンは10台のディスクモジュールと一台の制御モジュールを持つものとし，データベース内の総タプル数：検索タプル数：更新タプル数＝10000:100:10の比となる1000個のトランザクションを実行し，その総実行時間を測定した。その結果，ディスクモジュール１台と制御モジュール１台の場合（通常の同時実行制御のアーキテクチャに相当）に比較して，ほぼディスクモジュール台数に比例する約9.5 倍の高速化が確認された。

6．おわりに

並列関係データベースマシンＧＲＡＣＥに於ける同時実行制御について述べた。尚，ＧＲＡＣＥでは１トランザクション内の複数トランザクションステップを並列に実行することができる［１］。その為の拡張トランザクションモデル，及びアルゴリズム，実装方式等に関しては稿を改めて発表したい。

参考文献

［１］Fushimi,S. "System Software of A Relational Database Machine Based on Data Stream Model" Ph.D Thesis, Univ. of Tokyo (1986)

［２］Jordan,R., et al. "Precision Locks" Proc. of ACM SIGMOD Conf., pp.143-147 (1981)

［３］西村健 "並列データベースマシンに於ける同時実行制御" 東京大学卒業論文 (1986)