2F-5 可変構造多重処理データベースマシンの構成

喜連川優，赤松宏恒，田中英彦，元岡達
（東大　工学部）

§1 序　論

近年、Cellular Logic Typeのデータベースマシン（RAP，CASSM etc）の提案が多くなされてきたが、ここでは、それをより発展させた可変構造多重処理データベースマシンの構成、及びリレーショナルデータベースのサポートについて報告する。

§2 設計方針

1) 潜在的並列性の抽出-------- Cellular Typeでは、当該メモリにアクセス出来るプロセッサはそのメモリに付加されたプロセッサだけであり、従ってリレーションの占めるメモリセル数だけの並列度しか原理的に得られない構造になっている。メモリセルとは独立に任意台数のプロセッサを駆動できる環境を生成する事が好ましい。

2) プロセッサの有効利用--------Cellular Typeでは、自分のメモリに有効データがないとプロセッサはアイドルになる。セル数を増大させると、このコストは無視出来なくなり、プロセッサ、メモリのより柔軟な結合関係が望まれる。

3) マルチユーザ環境への適合性-------- SIMDマシンによるシリアルな処理は好ましくなく、多くのUser Queryがそれに適した数のプロセッサ群により並列に処理されるべきである。

§3 基本構成

前節の方針を満たすべくデータ操作部は、図1の如きリングバス上に多くのプロセッシングモジュール（PM），メモリモジュール(MM)，コントロールモジュール（CM）を結合した構成となっている。MMはデータを格納する磁気バブルメモリ，及びマーキング用半導体RAMから成り、データは必要なAttributeのみ選択、出力される。又不在データはステージングされる。PMはパタンマッチ処理を行ない、マーキング情報をMMに返す。リングバスは光ファイバにより高いデータ転送レートを確保し、これを固定数のチャネルに分割して利用する。CMがMM，PMにチャネルを割当てる事により任意の結合関係を実現できる。当該メモリに対し任意台数のプロセッサを駆動することが可能である。メモリからプロセッサ群へデータをブロードキャストすることにより、メモリアクセス競合のない環境を生成している。これによりプロセッサの駆動台数に略々比例した処理速度の向上が期待できる。メモリは当該オペレーションに必要なものだけが動的にリングバスへ結合される。

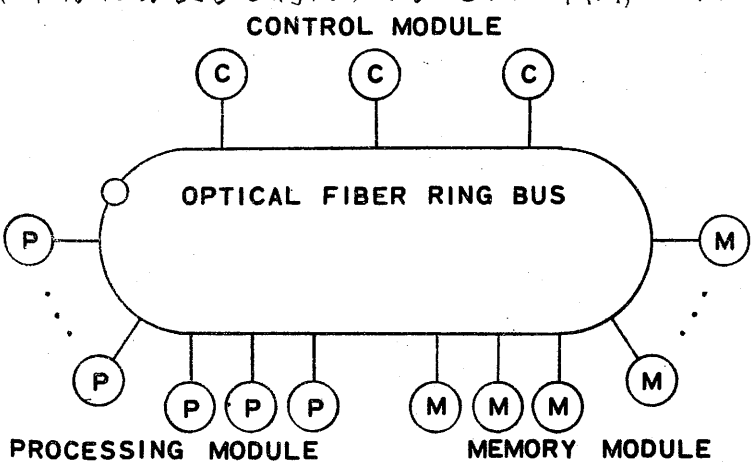

FIG. 1 ARCHITECTURE OF DATA MANIPULATION UNIT

§4 関係代数の処理方式

1) リングバスデータフォーマット--------メモリから送出されたデータに対してプロセッサ群は処理を施すが、この処理速度はバスのデータ転送レートではな

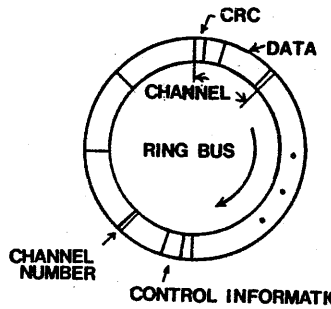

FIG.2 RING BUS DATA FORMAT

くメモリの出力レートによって定まる為プロセッサはその処理結果であるマーキング情報を当該データチャネルよりもいくつか後のチャネルに出力することになる。(図2参照)

2) 関係代数処理--------RESTRICTIONは Cellular Type と同様当該リレーションの占めるメモリモジュール数だけの並列性が得られる。JOINではプロセッサは一方のリレーションからタプルフェッチを行ない他方のリレーションを検索しマッチすると結果タプルを別のメモリに出力する。(IMPLICIT JOINではマーキング)従ってメモリ数とは無関係に、駆動プロセッサ数に比例した処理速度の向上が期待できる。R_i の Attribute A に関する R_j の Attribute B によるDIVISIONは、Divisor を $R_j = R_j^0 \cup R_j^1 \cup \cdots$ と分割すると $R_i[A \div B]R_j = (\cdots((R_i[A \div B]R_j^0)[\bar{A}=\bar{A}]R_i)[A \div B]R_j^1)[\bar{A}=\bar{A}]R_i)\cdots$ と表わされ、順次篩い落とし商を得るが、各段階を多くのプロセッサで並列処理する。

§5 Concurrency Levelの向上

1) マークビットとテンポラリリレーション--------マークビット処理方式ではその処理結果がホストマシンへ送出されるまで当該リレーションをロックする為並列度が低下する。ここではベースリレーションに対する選択に対しテンポラリリレーションを生成する事でベースリレーションを開放し、その後はマークビットにより処理をすすめるという中間策を採用する。

2) Concurrency Control

・Retrieval-Retrieval Concurrency------1)で改善されるが、よりアクセス頻度の高いものに対しては全タプルの送出により同時処理可能である。

・Retrieval-Update Concurrency------更新タプルはDifferential Setとして生成し、Retrieval Queryは、更新前のConsistentな値のみ検索することにより同時処理可能である。

・Update-Update Concurrency------Predicate Lockにより互いに干渉がないと判定された場合には各々個別にDifferential Setを生成する事により同時実行可能である。

§6 Semantic Integrityの維持

更新時には、ユーザによって指定されたAssertionを多数のプロセッサを用いて高速にチェックする事により Integrity を維持する。Tuple Assertionは更新時にチェックされ、Set Assertion(Aggregate Constraints, Functional Dependency etc)は更新後、Differential Set に対してチェックされる。Assertionを満たさない場合、マークビットの操作でDifferential Setを消去出来る為、Back Outに要する時間は僅かである。

§7 伝送制御手順

本システムでは、一台のメモリが多くのプロセッサ群に対してデータをブロードキャストするが、この際全てのプロセッサが正しくデータを受信した事を確認しつつ次データの送出を行なわねばならない。これに対してはメモリがマスタとなり障害プロセッサが回復(同期回復)するまで当該オペレーションを一時的に中断し、その後処理を再開する手法を採っている。プロセッサからメモリへデータを書き込む場合も同様に、メモリ、プロセッサいずれの障害も回復可能な伝送制御手順により正しいデータ送受が実現され、プロセッサ台数に依存した障害の影響を小さくしている。

§8 結び

現在、ハードウェアシミュレータによるバブルメモリモジュールの試作を終え、更にシステム全体の設計を進めている。