GRACEプロトタイプシステムにおける ソフトウェア構成

1B-7

中山 雅哉† 伏見 信也† 喜連川 優†† 田中 英彦† 元岡 達†

(†東京大学 工学部 ††東京大学 生産技術研究所)

1. はじめに

我々が研究・開発を続けている関係データベースマシンGRACEでは，複数のモジュールを並列に動作させることによって，データ流に沿った関係代数演算処理の実行を可能にしている〔1〕。現在実装を進めているGRACEプロトタイプシステム（以降，プロトタイプシステムと呼ぶ）では，各モジュールを汎用機上のソフトウェアプロセス（以下，プロセス）と必要なハードウェア資源との組に対応させることにより，本来のGRACEをシミュレートしている。本稿では，プロトタイプシステムにおける各プロセスの動作，全体的なプロセス構成等について報告する。

2. プロトタイプシステムのプロセス構成及び実装

本来のGRACEは，PM，MM，DM，WDM，CMの5つのモジュールを各々複数台，2つのネットワークで結合した構成をとっている。GRACEでは，それらのモジュールを並列に動作させることにより，データ流に沿った関係代数演算処理の実行を可能にしている。それに対して，本プロトタイプシステムでは，基本的な動作の為に必要な，最小限度のモジュール構成をとっており〔2〕，各モジュールは汎用機上のプロセスを用いて実装されている（Fig.1）。

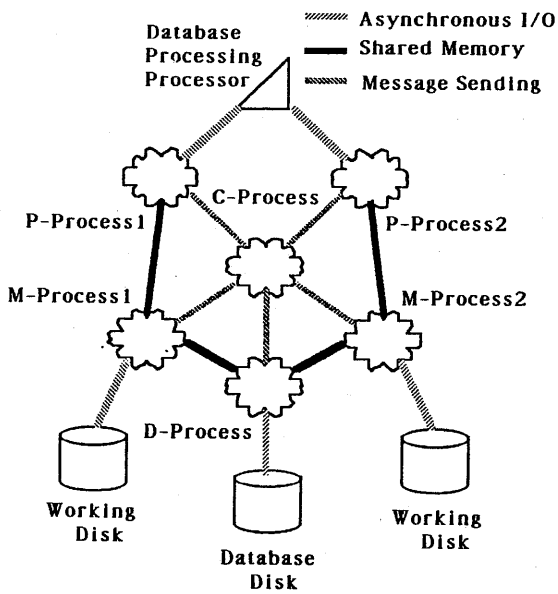

Fig.1 Process Architecture of GRACE Prototype System

この様な複数のプロセス構成によりデータ流指向のデータベース処理を効率良く実現する為には，次の点を考慮する必要がある。

(1) OSのサポートしている入出力は，通常システムが入出力用に集中管理する空間（システム入出力バッファ）を経て実行される。この様な入出力では，実際にユーザ空間にデータを読み込むためには，入出力本来のデータ転送に加えて，システム入出力バッファからユーザ領域への再度のデータ転送が必要となる。通常の処理環境下では，同一データに対する反復的な入出力が多い為，この手法は有効となるが，本プロトタイプシステムの様にデータ流指向の処理においては，一般にこの様な性質は期待できない。また，入出力を一つのプロセスが発行すると，当該プロセスは入出力完了待ちとなり，実行の制御は，他のプロセスに移されてしまう。この様な処理方式では，D－プロセス，WD－プロセスの様に，ディスクの入出力と並行してデータの処理を実行させる必要があるプロセスを，うまく実装することはできない。

(2) 仮想空間記憶管理や，プロセスの多重処理管理を実現しているOSでは，ユーザの要求とは別に，システム側でページ，プロセス等の単位によってディスクとの入出力を行うものがある。本プロトタイプシステムでは，D－プロセス，WD－プロセス，P－プロセスが，定常的に外部デバイスと入出力を行っている環境にある為，この様な，システムの実行する入出力は，システム全体の入出力負荷を上げることになり，好ましくない。

(3) 本プロトタイプシステムでは，Fig.1に示した様に複数のプロセスの間で，制御やデータの通信を行う必要がある。前者は少量の情報の授受で済むが，後者は，数kB以上の大量の情報を授受する必要がある。これを，UNIXのpipe機能のようにディスクを用いて通信したり，通常のメッセージ授受方式を用いたりする方法では，通信の為のオーバヘッドが大きくなってしまう。

この様に，OSのサポートする諸機能には，本プロトタイプシステムの実装において，かえってオーバヘッドを増すと予想されるものがある。本来ならば，プロトタイプシステム専用のOSを作成するのが好ましいが，今回は，入出力等に関して極力低レベルのOS機能を用いることにより，必要なプロセスの実装をすることにした。

以下に，上記の点に留意して構成したプロトタイプシステムのソフトウェア実装方法を示す。

(1)' 本プロトタイプシステムでは，今回実装に用いているＯＳ（ＤＰＳ１０）の提供する非同期入出力を使用している。この入出力は，ディスク内の指定したアドレスからのデータを，システムで使用している入出力バッファを経由しないで，直接プロセス内の任意の空間に転送する機能を有している。また本入出力を発行したプロセスは，入出力と並行して動作を継続することができる為，(1)の問題を解消することができる。即ち，ダブルバッファリングと本非同期入出力を用いることにより，大量のデータをデータ流として効率よく入出力することが可能となる。また，Ｐ－プロセスで用いているデータベース処理プロセッサ〔3〕は，Ｈｏｓｔとのインタフェースをディスクと同様になるように設計してあり，ここでもこの非同期入出力を使用することができる。

(2)' ＤＰＳ１０では，仮想空間記憶管理方式を採っていないので，システムによる自動的なページイン，ページアウトは起らない。また，このＯＳでは，プロセスを主記憶に常駐させる機能がある。本プロトタイプシステムにおけるプロセスは，各々比較的小規模なプログラムで実現可能であり，本機能を用いて全てのプロセスを主記憶に常駐させることにより，(2)の問題も解消することができる。

(3)' プロセス間の通信機能としては，(3)で述べた様に，制御とデータの通信では，その性質が異なっている。データの通信に関しては，大量データのプロセス間通信に適した主記憶の共有方式を用いている。また，制御の通信に関しては，通常のメッセージ授受方式を採っている。

3．各プロセスの動作概要

本節では，各々のプロセスの動作の概要について，説明する。

（ｉ）Ｃ－プロセス

本プロセスは，他のプロセスを制御し，データベース処理の実行を管理するプロセスで，ＧＲＡＣＥの処理手順に基づき，各関係代数演算に対し，処理に必要な資源の割当て，プロセスの初期化，及び起動を行う。

（ii）Ｄ－プロセス

Ｃ－プロセスにより起動されると，データベースの格納されているディスクからデータを読込み，指定された Filtering処理（ SELECTION演算等）を行って，Ｍ－プロセスにデータの送出を行う。ディスクからのデータの読込みと，Filtering処理，Ｍ－プロセスへのデータ送出は，前述の様に，並行して処理される。

（iii）Ｍ－プロセス

データ流のSource空間であるか，Sink空間であるかにより，処理が異なる。

(1)データ流のSink空間

Ｃ－プロセスにより，起動されると，ＷＤ－プロセスを駆動し，Sink空間の仮想化環境を設定する。Ｄ－プロセス又は，Ｐ－プロセスから転送されてくるデータにHash処理を行い，これらを複数のバケットから成るClustering空間として管理を行う。この空間がオーバフローした際は，適当なバケットを選択してＷＤ－プロセスに送出する。

(2)データ流のSource空間

プロセスが起動されたら，Ｐ－プロセスにバケット単位でデータを送出していく。また，これと重畳化してＷＤ－プロセスを駆動し，destageしたバケットを再stage する。これにより，Ｐ－プロセスに対して連続的にデータを供給することが，可能となる。

（iv）ＷＤ－プロセス

Ｍ－プロセスに格納できなかったデータを，バケットを単位として作業用のディスクに効率良く格納する操作を管理，実行する為のプロセスである。

（ｖ）Ｐ－プロセス

ソータを中心として，JOIN,AGGREGATION等の処理を行うデータベース処理プロセッサとのインタフェースを司るプロセスである。現在，試作しているデータベース処理プロセッサは，入力用と出力用の２つのポートを持っており，各々独立なＰ－プロセスにより，制御される。

4．あとがき

ＧＲＡＣＥプロトタイプシステムにおけるプロセスの動作概要を中心として，全体的なプロセスの構成及び，実装方法について報告した。現在，各プロセスのためのソフトウェアを実装中である。機会をかえて，各プロセスの詳細設計，本プロトタイプシステムの性能評価の結果について報告する予定である。

参考文献

〔1〕伏見他，『データベースマシンＧＲＡＣＥのアーキテクチャとその実行制御系』，アドバンスト・データベースシンポジウム（1984）

〔2〕伏見他，『データベースマシンＧＲＡＣＥのプロトタイプシステム』，情報処理学会第31回全国大会,1B-6（1985）

〔3〕鈴木他，『ＧＲＡＣＥプロトタイプシステムにおけるプロセッシグモジュールの設計』，情報処理学会第31回全国大会,1B-8 （1985）