3C-7

データベースマシンGRACEの制御ソフトウェア

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

並列データベースマシンの実装に於ては，マシン内の資源管理，資源間の協調制御，データ転送制御等を司る制御ソフトウェアは極めて重要である。特に，データベース処理に於ては，大量のデータが定常的に処理される為，ページ等の比較的小さなgranule を単位としてマシンの制御を行うと，処理単位に対する制御オーバヘッドが処理全体を通じて蓄積し，全体の処理時間に対して支配的となる。本稿では，オペランドデータ全体を制御単位とし，その流れに沿って処理を行うデータストリームモデルを提案し，本モデルに基づく効率の良い並列関係データベースマシンの制御ソフトウェアの構成，実装について考察する。

2.並列データベースマシンに於ける制御ソフトウェア

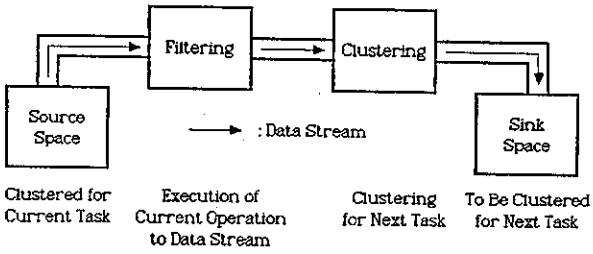
以下，並列関係データベースマシンDIRECTの試作機による性能評価結果〔1〕を基に並列データベースマシンに於ける制御ソフトウェアについて考える。DIRECT試作機に於ては，処理そのものよりも，その制御に要する時間が全体の処理時間の大半を占めることが明らかとなった。

一般に，並列データベースマシンに於ては，プロセッサの処理データ要求，処理データの転送，処理データに対する演算実行，結果データの格納空間の要求，結果データの格納，の繰返しが処理の基本となり，マシンの制御モジュールがこれら動作間の遷移時に介入することによって複数演算の並列実行が実現される。DIRECTはこれをページ（4Kバイト）を単位として実現している為，制御モジュールは処理または結果ページをプロセッサが要求する度に頻繁に呼ばれる結果となる。これらページを得る為には，制御モジュールは管理情報に対する比較的複雑な検索動作を行う必要があり，結果として制御モジュールがマシンの隘路となる。また，オペランドデータ，結果データはページを単位としてマシン内を転送されることになり，単位データに対するデータ転送オーバヘッドも増加することになる。

3.データストリームモデル

上記の問題を解決する為には，(1)処理・制御の単位をページ等の比較的小さいものから，演算のオペランドデータ全体に拡大する，(2)オペランドデータの転送と処理を同時に行う，所謂データストリーム処理を導入する，ことによって解決される。我々は，この考えに基づくデータベース処理のモデルをデータストリームモデルとして定式化した。

データストリームモデルに於ては，処理データを格納するSource空間，データストリームに対し，その流れに沿って演算を実行するFiltering，結果データストリームに対し，次演算に対する動的クラスタリングを実行する Clustering，クラスタリングされたデータストリームを格納するSink空間が1演算の実行環境を形成する（図1）。デー



Figl. Structure of Execution Environment of Task

タストリームがその実行環境中を流れ，演算の実行を行うことをタスクと呼ぶ。タスクはマシンの制御に於て最小の単位であり，その発火（タスク内のデータストリームの送出開始）後はマシンの制御モジュールからの指示なしで自律的に動作する。タスクは木構造にまとめられて，トランザクションステップを形成する。トランザクションステップはＳＱＬ等の1statement の実行に対応する。更に，トランザクションとはアプリケーションプログラム中の依存関係とconsistent，且つ，serializableなトランザクションステップの並列実行と定義される。

4.制御ソフトウェアの構造

データストリームモデルに基づく制御ソフトウェアは4層の階層構造により実現される。最下層は基本アルゴリズム層であり，種々の演算に対する具体的な処理アルゴリズム，即ち，タスク内のFiltering，Clustering を実行される演算に応じて実現する。第2層はタスク内制御層であり，タスクの発火後，タスクの実行を終了するまでの間，タスク内のデータ転送制御を行う。この層はタスク毎に閉じたソフトウェア層であり，マシンの制御モジュールとは独立に実現され，タスクに自律性を与える。第3層はタスク間制御層であり，タスクに対する資源の割付け・発火・資源の開放を司ると共に，更新演算を含むタスク間の同期制御を行う。トランザクションステップ内のタスクの木構造は，本層に於て兄弟タスクがSink空間を共有する様，資源を割付けることにより実現される。第4層はトランザクション

管理層であり，トランザクション内の実行可能なトランザクションステップを選択し，それを木構造にコンパイルしてタスク間制御層に転送する。また，トランザクションのコミット制御，障害回復制御を司る。

5.GRACE に於ける実装

我々は，上記の制御ソフトウェアの構成方式を並列関係データベースマシンGRACE に適用し，その実装について検討した。GRACE のハードウェアアーキテクチャを図2に示

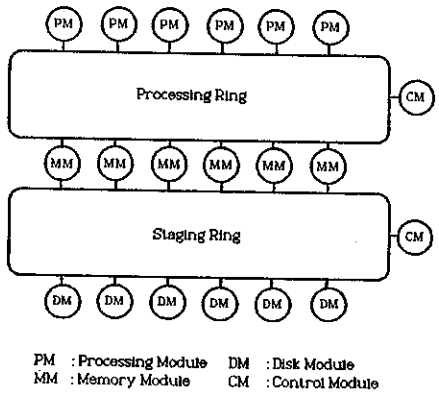


Fig.2 Hardware Organization of GRACE

す。GRACE に於ては，上記制御ソフトウェアの第1層は，プロセッシングモジュールに於けるハードウェアソートを基本としたjoin等の処理アルゴリズム，ディスクモジュールに於けるselection 等のアルゴリズム，Hashを用いた動的クラスタリングアルゴリズム等〔2〕として各モジュール内に実装される。第2層は，各モジュールに付加されたリングバスインタフェースユニット間で用いられるタスク内のデータストリームプロトコルとして実現される〔3〕。第3層はリングバス上に存在する2つの制御モジュール上に実現される。第4層は下部リング上の制御モジュールとホスト計算機上に置かれる。ここでは，以下第3層の実現を中心として，join演算の実行を例にとり，GRACE に於ける制御ソフトウェアの実装について述べる。

GRACE に於けるjoin演算は，joinされるタプルを格納しているメモリモジュール群（Source空間）からプロセッシングモジュール（Filtering 及びClustering）を経て結果タプルを格納するメモリモジュール群（Sink空間）へ向かってデータストリームを流すことにより実行される。join演算に対応するタスクが，タスクの木構造に於て発火可能（その子タスクが全て実行を終了した）になると，タスク間制御層の資源管理はリングバス上の資源管理チャネルにSource空間を構成するメモリモジュール，プロセッシングモジュール，Sink空間を構成するメモリモジュールの台数を各々書き，実行しようとするタスクのIdと共にリングバス上に送り出す。一方，各モジュールは，当該タスクのデータを格納している（Sourceメモリモジュール）か，Idle（プロセッシングモジュール，Sinkメモリモジュール）であれば，送られてきた資源管理チャネルの対応する台数のフィールドをデクリメントし，資源管理からの応答を待つ。資源管理は，リングバス上を一周して戻ってきた資源管理チャネルの台数フィールドの値によって必要な数の資源の確保の成否を判断することができる。この際，Sinkメモリモジュールが期待した台数確保できない場合でも，Sink空間の仮想空間化〔3〕によりタスクの実行を開始することができる。資源管理は，タスクの実行の判断結果に従ってack+初期化コマンド，またはnackを書いて，資源管理チャネルを再びリングバス上に送り出す。各モジュールは，資源管理からの応答に基づき，初期化等の動作を行う。

この様にして必要なモジュールが割付けられたタスクはタスク内のデータ転送を行う為のデータ転送チャネルを割付けることにより実際に発火される。発火と同時に，タスク内ではデータストリーム制御プロトコルが起動され，制御モジュールの介入なしでデータ転送，演算の実行が行われる。一方，タスクの実行状態はリングバス上を流れるタスク制御チャネルにより監視される。本チャネルは，資源管理チャネルと同様に3つのモジュール台数フィールドを持つ。タスク内で実行を終了したモジュールは，対応する台数フィールドをデクリメントする。タスク間制御層の資源管理は，これらフィールドが全て0となるとタスク内のデータストリーム処理が終了したと判断し，タスク内の資源の開放を行う。

以上の様に，本構成方式に基づくGRACE の制御ソフトウェアの実装に於ては，マシンの制御モジュールは，タスクの実行開始，及び終了時のみに呼び出され，演算自体はタスクとして制御の介入なしに自律的に実行される。従って，上記の2つの問題点は根本的に解決されており，極めて小さなオーバヘッドで並列データベースマシンを効率良く制御することが可能となる。

6.おわりに

データストリームモデルに基づく並列データベースマシンの制御ソフトウェアの構成方式と，GRACE に於けるその実現について述べた。本方式に於ける同時実行制御，障害回復管理等については稿を改めて報告したい。

参考文献

〔1〕Boral,H., et.al.『Implementation of the Database Machine DIRECT』Trans. on Soft. Eng., SE-8(6), 1982

〔2〕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

〔3〕伏見他『リングバスを用いたGRACE のモジュール間結合系』情報処理学会第29回全国大会，3F-7，(1984)