拡張関係データベースエンジン XRDBのシステム・アーキテクチャ

6Q-6

山根 康男*, 成田 実香**, 小櫻 文彦**
* ㈱富士通研究所, ** 富士通株式会社

1. はじめに

データベースで扱うデータは従来の文字，数値からマルチメディア（テキスト，画像，等）や知識というように拡がってきている。また，アプリケーションも知識処理やCAD と言うように多様化してきている。一方，関係データベースは遅いと言われ，データベース・マシン等の研究が盛んに行われてきたが，知識やマルチメディア等の情報処理の高度化・扱うデータの大量化に伴い，ますます高速性が要請されている。これらの情勢に鑑み，我々はＮＦ² 〔１〕を実現するよう機能拡張され，また高速性を追求した拡張関係データベースエンジンXRDB（eXtended Relational DataBase engine)を開発した。

関係データベースシステムは一般に，問い合わせ言語の解析部，最適化部，実際に処理を行う実行部の３つにおおまかに分けることができる。XRDBはこの中で実行部にあたるシステムである。XRDBを開発した大きな目的は，関係データベース処理を高速に行うことである。このために，従来ソートが使われていた処理にハッシュを適用する等，色々な高速化の手法を適用した。

また，今後ますます多様化すると思われるアプリケーションやデータに対処するために，機能拡張のことも十分に考慮に入れ設計を行った。

2. 設計方針

以下，XRDBの基本的な設計方針について述べる。

(1) 単純性

単純でコンパクトなシステムを作る。単純なシステムは保守も容易であるし，性能的にも高速で質の高いプログラムを作りやすい。

機能的には，プリミティブなデータベース本来の機能に留める。前にも述べたように，アプリケーションの多様化に伴い，最適化に関してもカスタマイズしたいという要求がある。この意味でも，XRDBでは最適化をユーザに任せ，実行部分のみを提供することにした。

XRDBでは辞書，インデックスという概念はない。これらはXRDBで提供するリレーションを使ってユーザに実現してもらう。例えば，インデックスであれば，後述のB-treeリレーションにより実現してもらう。即ち，XRDBでは，扱うデータは，全てリレーションとして格納される。

(2) 拡張性

データベースの処理を考えてみると，検索条件の評価のようにアプリケーションごとに異なるアプリケーションに近い処理がある。一方，検索，挿入，削除や，排他制御，トランザクション管理のような処理はアプリケーションに共通したデータベース本来の機能と呼ぶべきアプリケーションに共通な処理である。例えば，従業員のリレーションの中から年令が30才の人の給料の平均値を計算すると言うアプリケーションでは，年令が30であるかどうかの条件判定や，給料の平均値を求めるという処理はアプリケーションに近い処理である。

アプリケーションに近い処理に関しては，機能拡張の要求が強い。例えば，倍精度の計算をしたいとか，構造をもったデータを扱いたい等である。また，システムでサポートしていない機能やデータ型をユーザ定義関数やユーザ定義データとして定義したいという要求がある。この機能拡張の要求はアプリケーションの多様化に伴い，ますます強まるものと思われる。

XRDBの設計の基本方針の一つは，アプリケーションに共通した処理を高速に行うシステムを作るということである。アプリケーションに近い処理はユーザ定義関数として渡してもらう。また，データ型に関しても，XRDBでは固定長，可変長のみサポートする。したがって，タップル識別子をフィールド値とすることも可能である。データの型は，ユーザにユーザ定義関数の中で定義してもらう。

XRDBで，ユーザ定義関数としたものは，検索条件を示す述語，検索条件に適合したタップル毎に行う処理，ソートやB-treeにおいて順序を指定する順序関数，B-treeにおける検索範囲を示す関数，およびハッシュ関数である。

このようにXRDBでは，アプリケーションに近い処理やデータ型をユーザ定義としているが，これらの機能は将来的にも，拡張要求の強い部分であり，これらの部分

The system architecture of the extended relational database engine XRDB
Yasuo Yamane[1], Narita Mika[2], Fumihiko Kozakura[2]
1)FUJITSU LABORATORIES LTD., 2)FUJITSU LIMITED

をXRDBから切り離すことは，モジュール性，拡張性という点からもよいと考えている。

3. 高速化のための手法

以下XRDBで採った高速化のための手法について述べる。

(1) ハッシュの活用

等結合，重複除去，差集合等，従来ソートを用いて行っていた処理をハッシュを用いて行う。このことにより，オーダが O(n log n) からO(n)に改善される〔2〕。

(2) 処理，データ構造の単純化

実現に際して，処理，データ構造は極力単純化した。タップルは極力バッファ上から動かさないという方針で設計した。例えば，ソートでは，タップルを並べかえるかわりにタップルへのポインタの並べかえにより行う。また，フィールドへのアクセスはバッファ上のタップル内のフィールドへのポインタにより行う。ユーザ定義関数への値の受け渡しもこのポインタにより行う。

(3) ユーザ定義関数による高速化

述語の評価等はインタプリタとして作成すると重い処理となり，また一般に機能を増やせば増やすほど効率が悪くなる部分である。。XRDBではこれらの処理を最適化されたユーザ定義関数として渡してもらうことにより，高速な述語評価等を可能にする枠組みを提供している。

4. システム・アーキテクチャ

ここでは，XRDBのシステム・アーキテクチャについて説明する。まず，XRDBでサポートするリレーションは以下の4 種類である。

(1) シーケンシャル・リレーション　挿入された順にタップルが格納される。ページはシーケンシャルにリンクされる。

(2) B-treeリレーション　B-treeの構造をもつリレーションであり，リーフにタップルが指定された順序に従って格納される。

(3) ダイナミックハッシュ・リレーション　Larsonのpartial expansion〔3〕に基づくダイナミック・ハッシュ構造を持つリレーションである。

(4) 内部リレーション　NF^2 を実現するためのフィールド値としてのリレーションである。内部的には可変長フィールドとして実現されている。

XRDBは図1に示すようにUNIX OS の上に，関係演算，タップル，ストレージの三階層からなる。XRDBは効率，移植性を考慮してC 言語で実現されているが，コード量は約15キロステップである。

関係演算レイヤは選択，結合，ソート，ネスト(FNFからNF^2 への変換)，アンネスト(NF^2からFNF への変換)等の拡張された関係演算を提供する。全体で15の関係演算をサポートする。

タップル・レイヤは，前述の4種類の各リレーションに対する，タップルの検索，挿入，削除，更新等をサポートする。

ストレージ・レイヤはページ単位のバッファ管理をサポートするレイヤであり，トランザクション管理も行う。NF^2 やマルチメディアの格納を考慮して，ページの大きさは従来の4KB だけではなく，8KB，16KB，32KB，64KB をサポートした。

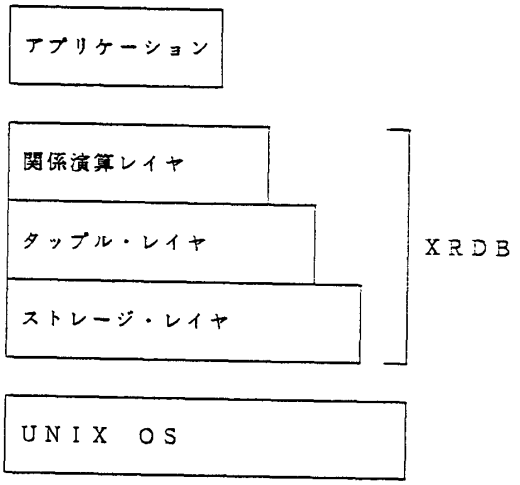

図1　XRDBのシステム構成

5. まとめ

現在，シスグル・ユーザ版が完成し，性能評価を行った。今後はリカバリ機能および排他制御機能を追加し，マルチ・ユーザ化する予定である。

謝辞

この研究は通産省工業技術院大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」の一環として行った。

参考文献

〔1〕 Makinouchi, A., A consideration on Normal Form of Not-necessarily-Normalized Relations in the Relational Data Model, Proc. 3th Conf. VLDB, (1977), pp.447-453.

〔2〕 Yamane,Y., A hash join technique for relational database systems, Proc. Int. Conf. on Foundations of Data Organization, (1984), pp. 388-398.

〔3〕 Larson, P.A., Linear hashing with partial expansions, Proc. 6th Conf. VLDB, (1980), pp 224-232.