

## 高速フィルタリングプロセッサの方式と性能評価

土田正士 河村信男 中野幸生 武藤英男 北嶋弘行 米田 茂  
(株) 日立製作所システム開発研究所

高速フィルタリングプロセッサの実現方式と性能評価結果について述べる。本プロセッサは、メインフレーム計算機のチャンネルに接続された付加プロセッサの位置付けにある。対話処理の性能向上とオンライン処理への影響を最小化することを目指し、本プロセッサを支援するプロトタイプDBMSを開発した。性能評価では、対話処理での性能実測を定量的に評価した結果を示す。

## A Highspeed Filtering Processor Architecture and Its Evaluation

Masashi Tsuchida Nobuo Kawamura Yukio Nakano Hideo Mutoh  
Hiroyuki Kitajima Shigeru Yoneda  
Systems Development Laboratory, HITACHI Ltd.  
1099 Ohzenji, Asao-ku, Kawasaki 215, Japan

A highspeed filtering processor architecture and its evaluation are described. This processor is a filtering processor attached to a conventional channel of a mainframe computer. We extend the commercial DBMS for cooperating filtering processor capability. It is designed to improve the interactive query performance and minimize the influence on on-line transaction response time. The measurement results of the extended DBMS, in which interactive queries are concurrently executed, are described and analyzed.

## 1. まえがき

関係データベースは、意思決定支援システム、文書検索システムを始めとして対話的に処理を行う非定型業務でのニーズが高い。これらのシステムでは、インデックス手法を用いて処理の高速化を図っていた。しかし、インデックス手法だけで高速化を達成するには以下のような弊害があり、性能上の問題となっている。

(1)条件で指定されるカラムにインデックスが存在するとは限らない。過度にインデックスを張ることは、更新処理の性能劣化を招く。

(2)インデックスが付いていても、統計処理を始め、インデックスを利用した処理によって過大な入出力が発生する場合がある。

(3)文書検索システムでは、インデックスを利用してもキーワードの陳腐化や分解能不足の問題があるため、自由キーワードによる全文検索を併用せざるを得ないことがある。

関係データベースシステムの高速化の一手段として、データベースマシンの開発が行われてきた[1][7]。しかし、いくつかの問題が残されている。実用システムでは、ユーザの抱えるデータベースが大規模になっている。そのため、少なくとも近い将来に主記憶上に全てのデータベースを常駐化することは不可能であり、2次記憶装置上のデータベースを高速にアクセスする手段が必須である。また、一般的に複数ユーザが同時にデータベースをアクセスする環境では、オンライン処理と対話処理が混在し、並列に実行される。特に、オンライン処理はユーザのアプリケーションシステムの中で最も重要なサービスであり、迅速な応答性能が要求される。そのため、2次記憶上のデータベースを処理対象とするデータベースマシンが稼働する環境において、オン

ライン処理と対話処理が混在した場合、オンライン処理への影響を評価する必要がある。

本稿では、高速フィルタリングプロセッサの実現方式と性能評価について述べる。2章では、解決すべき課題について述べる。3章では、システムの概要について示す。4章では、プロトタイプDBMS（以下、プロトタイプと略記する）の機能概要について述べる。5章では、性能評価結果とその考察を述べる。

## 2. 課題

### (1)オンライン処理への影響

対話処理とオンライン処理が並列に実行される環境において、データベースマシンを用い対話処理の応答時間を短縮することが可能である。データベースマシンの性能を最大限に発揮させるためには、大きな排他制御の単位、あるいは入出力装置の占有時間を長くするため、多くのバッファ資源とページロック資源が必要となる。このような実装方式では、オンライン処理の性能低下を招く。この問題を解決するためには、バッファ資源の最適配置方法とページロックの早期開放方法が必要となる。本機能の実現により、オンライン処理、対話処理双方での性能向上が期待できる。

### (2)格納構造への対応

多くのデータベースマシンは、特定の格納構造にだけ対応していた。メインフレーム計算機システムでは、異なった格納構造を持つ複数のデータベース管理システムが稼働している。そのような環境を想定して、高速フィルタリングプロセッサ（以下、DBマシンと略記する）の接続インターフェースは、複数の格納構造に対応できなければならない。

### 3. 概要

DBマシンは、チャンネルと入出力制御装置との間に位置し、入出力装置からのデータ転送中に選択、文字列検索、射影、計数等のデータベース演算を実行する。これによって、CPU負荷を大幅に軽減し、アドホックな問合せの応答性能を向上させる。

#### 3.1 システム構成

図1は、プロトタイプシステムの構成である。DBマシンは、チャンネルインタフェースを介して、チャンネルと入出力制御装置との間でデータの転送を制御し、データベース演算（選択、射影）、文字列検索機能を実装する専用のハードウェアである。このプロトタイプの特徴を示す。

##### (1) データ転送への追従

複雑な条件式に対する処理要求でも、今後一層高速化するデータ転送速度に追従可能とするため、専用ハードウェアをバイプライン動作させる。

##### (2) ソフトウェアによる高速化処理

入出力発行、排他制御、バッファ共有制御、処理結果の受け渡しを一括処理することによって、ソフトウェアによるボトルネックを発生させない。

##### (3) システム構成の透過性

DBマシンの導入により、アプリケーションプログラム、データベース構造に影響を与えないようにする。DBマシンを利用するか否かは最適化処理で自動的に判断する。

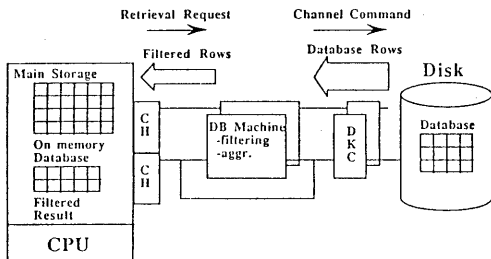


図1. システム構成

#### (4) システム構成の拡張性

DBマシンは、チャンネルと入出力インタフェースで接続されるので、各種CPU、入出力装置に接続可能である。そのため、負荷の増加に対しては、DBマシンの増設で対処し、また並列動作も可能とする。

#### 3.2 データベースマシンインタフェース

図2は、チャンネルとDBマシン間のインタフェースである。このインタフェースは、チャンネルコマンドを拡張し、設定した。これらコマンドの機能を以下に示す。

##### (1) SLI (Set Length Information)

SLIは、以下のSDB、SOD、SADコマンドの長さを指定する。

##### (2) SDB (Set Database)

SDBは、DBマシンにデータベースの格納構造を知らせる。この構造情報で、様々なレコード形式を処理可能とする。

##### (3) SAD (Set Address)

SADは、読み込むべきディスク上の格納範囲を知らせる物理アドレス列とからなる。

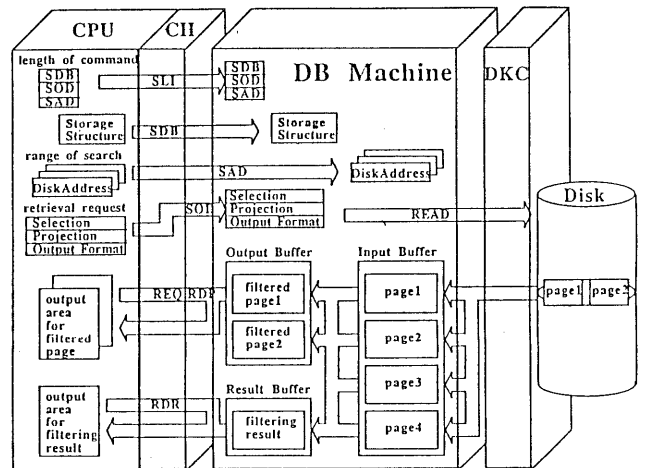


図2. データベースマシンインタフェース

#### (4) S O D (Set Order)

S O Dは、条件式、射影カラム、フィルタリングレベルからなる。フィルタリングレベルは、次の2つである。

—カラムレベル—条件式を満足するロウをページ形式に編集する。

—ページレベル—アクセスした全てのページを転送する。

D Bマシンは、S O Dコマンドを解析して、一連のチャンネルコマンドを入出力装置に発行する。S A Dコマンドによりディスク上の物理アドレス列として通知されたページは、入力バッファに読み込まれる。同時に、D Bマシンは既に読み込んであったページを検索して、出力バッファにフィルタリングレベルに合わせて編集する。また、ページ出力処理もこれら処理と並行して行われる。さらに、結果情報として、条件式を満足するロウを含まないページ、およびロウがオーバフローしており条件式の

評価ができない等の情報を蓄積する。

#### (4) R E Q (REquest Page)

R E Qは、D Bマシンが検索結果を転送可能か否かをチェックする。

#### (5) R D P (Read Page)

R D Pは、出力バッファに格納、あるいは編集されたページを読み込む。

#### (6) R D R (ReaD Result)

R D Rは、S O Dコマンドで蓄積された結果情報を読み込む。

### 4. ソフトウェア構造

図3は、プロトタイプソフトウェア構造である。プロトタイプは、商用D B M Sを機能拡張することによって実装した。機能追加した部分についての特徴を示す。

#### 4.1 データ操作

商用D B M Sでは、物理構造として固定長ブロックからなるページを採用している。表のロウは、ページに分割、格納される。さらにロウは、同一ページの連続領域に格納される。データベース操作では、ロウが処理単位である。即ち、各アプリケーションプログラムは、一時に一口ウの単位で処理を行う。ディスク入出力もランダムにロウが格納されるページ単位で発行される。したがって、ロウが商用D B M Sでは処理の単位であった。

プロトタイプでは、データ操作が一時に複数ロウを処理可能とし、排他制御、及びページアドレス変換のオーバヘッドを最小化するように設計した。D Bマシン要求は、予めバッファ上に存在するページを処理対象からはずし、複数ページ単位で発行される。排他制御は、一回のディスク入力アクセスされる複数ページの単位で行われる。D Bマシン要求の終了後、条件式を

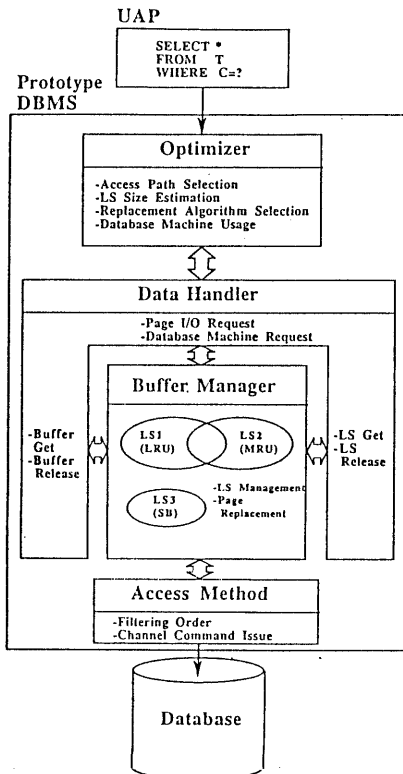


図3. ソフトウェア構造

満足しないロウだけを含むページに対するロックは、DBマシンの結果情報を基にして開放される。この情報を用いて、プロトタイプタイプのロック占有時間を最小化する。

## 4.2 バッファ管理

バッファ管理は、ページスラッシングを防ぐ目的でDBMIN方式を基にした[8]。ただし、オンライン処理、バッチ処理等に対応するためにバッファをグループ化する環境に合わせるようにDBMIN方式を変更した。DBマシンは、一回のディスク入力で複数ページをアクセスし、バッファプールに編集ページを転送する。この編集ページは、アプリケーションプログラムによって取り出される中間結果としてだけでなく、引き続くデータベース操作でアクセスされる内部データベースのページとしても扱われる。

バッファ管理は、バッファプールを複数のサブプールに分割する。このサブプールを、LRU、MRU等の置換アルゴリズムで管理されるLS(Locality Set)と呼ぶ。DBマシン要求は、複数ページを連続的に置き換えるSB(Single Buffer)アルゴリズムを適用する。これによって、オンライン処理に対する影響を最小化する。

## 4.3 最適化

最適化は、SQL[9]コマンドの解析を行い、LSサイズ、置換アルゴリズム、DBマシン利用可否からなる処理手順を決定する。最適化は、CPU負荷、及び入出力回数をデータベースの統計情報を利用して見積る[10][11]。DBマシン利用の基準は、利用できるインデクスの存在とCPU、及び入出力性能に依存する。

## 4.4 アクセス法

アクセス法は、DBマシンをアクセスする専用の入出力プログラムである。アクセス法は、4つのパラメタを解釈し、DBマシンをアクセスするチャンネルコマンドに変換する。次のパラメタからなる。

### (1)検索オーダ

このパラメタは、条件式、射影カラム、出力形式、フィルタリングレベルを指定する。検索オーダは、SODコマンドに変換される。

### (2)RBAリスト

入出力装置は、VSAMデータセットで管理される。各データセットは、RBA(Relative Byte Address)で位置付けされる。RBAリストは、DBマシンが読み込むページを指定する。RBAは、VSAMのディレクトリ情報を用いてディスク上の物理アドレスに変換される。各物理アドレスは、シーク、サーチ時間を最小化するため、昇順に並び換えられる。

### (3)編集ページアドレスリスト

これは、出力形式で指定された情報に基づき編集されたページを格納する領域を示す。編集ページアドレスリストから、REQ、RDPからなる一連のコマンドが作成される。

### (4)検索結果アドレス

これは、オーバフローロウ情報とアンロックページ情報とからなる。

## 5. 性能評価

プロトタイプの性能評価結果を示す。先に、シミュレータによって応答時間、及びCPU利用率の観点で評価を行っており、それらの結果を実機で検証する目的もある[12]。性能評価は、2段階に分けて実施した。最初に、対話処理が混在する環境での

応答時間、及びCPU利用率を評価した。次に、プロトタイプの総合的な評価について述べる。

### 5.1 評価モデル

実機評価の目的は、複数のユーザが並列にアクセスするデータベース環境での性能特性を把握することである。特に、対話処理とオンライン処理が混在する環境での評価は、データベースシステムに対するユーザの要件からみて重要である。図4は、対話処理が混在するシステムでの評価モデルである。このモデルでは、各処理でアクセスする表の配置、及び共有の有無をパラメータとした。対話処理のユーザ数を1から3まで変化させ、各ユーザが同一の入出力装置に存在する表を共有する場合、各ユーザが異なる入出力装置の別々の表をアクセスする場合について評価する。

### 5.2 対話処理の混在評価

図5、6は、対話処理が混在する環境での性能評価結果を示す。x軸は、応答時間比、CPU利用率、y軸は、対話処理ユーザ数である。まず、各々の応答性能を見ると、同一入出力装置上の表に対する対話処理では、約1/6から1/3に応答時間が短縮され、ユーザ数が増加するとDBマシン導入の効果が大きいことが分かる。一方、異なる入出力装置上の表に対する対話処理でも、応答性能は約1/3に短縮されており、DBマシンの有効性を示している。これは、複数ページを一括して処理することによる入力処理時間の削減効果とDBマシン適用によるCPU負荷の削減効果による併用効果に起因する。

ここで、各環境におけるCPU利用率の観点での考察を行う。同一入出力装置上の表に対する対話処理において、既存のD

BMSでは同一の入出力装置に対する入力処理要求に対して並行処理ができないため、ユーザ数が増加するに従い入出力処理のネックのため、応答性能が劣化する。CPU利用率を見ても分かるように、ユーザ数が増加すると、逆にCPU利用率が下がっており、入出力処理待ちが多発している。DBマシンを適用する場合、CPU利用率は、

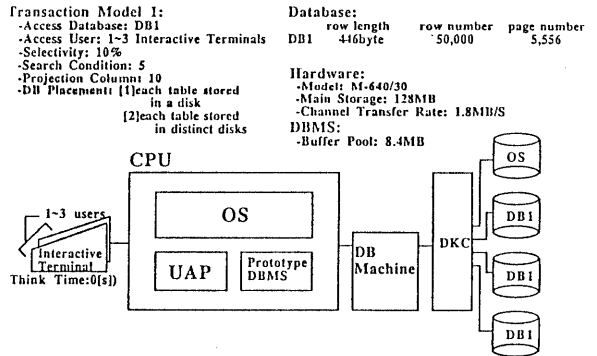


図4. 評価モデル

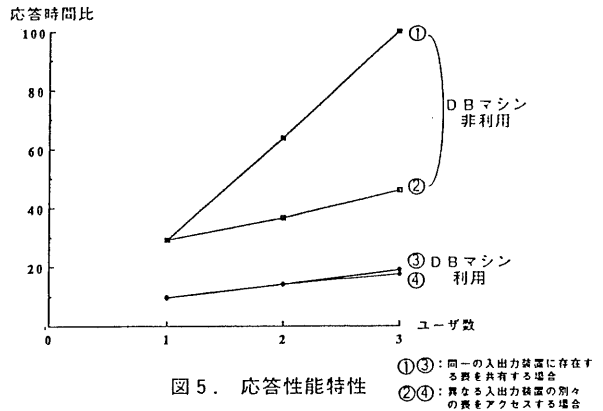


図5. 応答性能特性

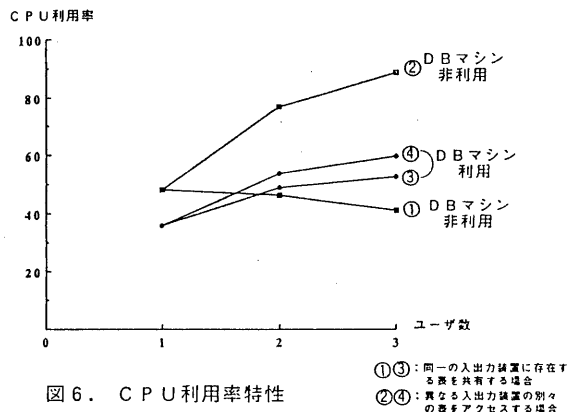


図6. CPU利用率特性

50%程度でありまだ余裕がある。また、異なる入出力装置上の表に対する対話処理において、応答性能がほぼリニアに伸びているのに対して、CPU利用率は、DBマシンを適用する場合にはユーザ数が3までリニアに伸びていない。これは、DBマシンが同一チャンネルに対する検索要求をシリアル化し、1つの入出力装置の処理中には他の入出力装置に対する処理を遅らせていることに起因する。この影響は、オンライン処理との共存において、オンライン処理が遅れる現象として現われる可能性がある。

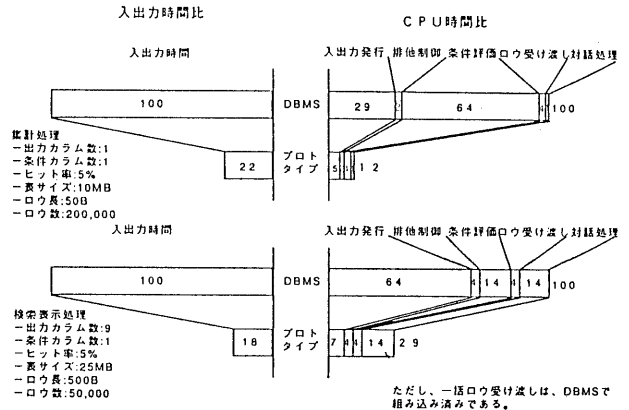


図7. プロトタイプの削減効果

### 5.3 プロトタイプの評価

プロトタイプで採用した各処理方式の評価を述べる。

入出力発行、排他制御、バッファ共有制御、処理結果の受け渡しを一括処理することによって、ソフトウェアによるボトルネックを防ぐことができた。図7は、各処理部分の削減効果を示している。図によれば、入出力発行処理の削減効果が顕著である。また、DBマシンの検索結果情報を用いて、条件式を満足しないロックを含むページに対するロックの占有時間を最小化する効果は、本評価では明らかでない。さらに、バッファ共有制御に関しては、一部バッファの効果により入力処理時間が短縮しているが、オンライン処理への影響を軽減しているとの観点での効果が見えていな

い。これら方式の評価は、より高負荷環境での実測で検証する必要がある。

### 6. まとめ

高速フィルタリングプロセッサDBマシンの実現方式と性能評価結果について述べた。プロトタイプDBMSの開発に際して、対話処理の性能向上とオンライン処理の応答時間への影響を最小化することを目指した。そのため、入出力発行、排他制御、バッファ共有制御、処理結果の受け渡しを一括処理する方式を採用した。それによって、既存の処理に影響を与えずに、ソフトウェアによるボトルネックを防ぐことができた。性能評価では、対話処理の性能向上を性能実測にて定量的に評価した結果を示した。

## 参考文献

- [1]Babb E.,"Implementation a Relational Database by Means of Specialized Hardware", ACM TODS, Vol.4, No.1, March 1979, pp.1-29.
- [2]DeWitt D.J., Gerber R.H., Graefe G., Heytens M.L., Kumer K.B., and Muralikrishna M.,"GAMMA-A High Performance Dataflow Database Machine", Proc. of 12th VLDB, 1986, pp.228-237.
- [3]Fushimi S., Kitsuregawa M., and Tanaka H.,"An Overview of the System of a Parallel Relational Database Machine GRACE",Proc. of 12th VLDB, 1986, pp.209-219.
- [4]Kojima K.,Torii S., and Yosizumi S.,"IDP-A Main Storage Based Vector Database Processor", Database Machine and Knowledge Base Machine, Kluwer Academic, 1988, pp.237-250.
- [5]Inoue U., Hayami H., Fukuoka H., and Suzuki K.,"RINDA-A Relational Database Processor for Non-Indexed Queries", Proc. of DASFAA '89, pp.382-386.
- [6]佐藤,中村,吉田,熊谷他 : 「データベース処理のための専用プロセッサ」,情報処理学会論文誌, Vol.23, No.4, 1982,pp.349-357.
- [7]北嶋,大曾根,山本 : 「高速フィルタリングプロセッサ実験システムの開発(1) -全体構想-」,情報処理学会第38回全国大会, pp.956-957.
- [8]Chou H.T. and DeWitt D.J.,"An Evaluation of Buffer Management Strategies for Relational Database Systems, Proc. of 11th VLDB, 1985, pp.127-141.
- [9]ISO/IEC 9075: 1989(E) - Database Language SQL
- [10]Sato K., Tsuchida M., Nakamura F., and Oomachi K.,"Local and Global Query Optimization Mechanisms for Relational Databases",Proc. of 11th VLDB, 1985,pp.405-417.
- [11]土田,酒井,大町 : 「DBマシンを含めたDBMSの最適化方式の提案」,情報処理学会第34回全国大会, pp.956-957.
- [12]酒井,河村,土田,武藤 : 「高速フィルタリングプロセッサ実験システムの開発(4)-性能評価-」,情報処理学会第38回全国大会, pp.962-963.