解　説

データベースプロセッサ

内蔵データベースプロセッサ IDP とフィルタリングプロセッサ RDSP の概要†

土田正士††　鳥居俊一††

1. まえがき

リレーショナルデータベースは，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） 条件では指定されるカラムにインデクスが存在するとは限らない．過度にインデクスを張ることは，更新処理時にインデクス保守のための性能劣化を招く．

（2） インデクスが付いていても，検索範囲が絞り込めない統計処理を始め，インデ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，インデクスを利用してもキーワードの陳腐化や絞り込み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えないことがある．

また，リレーショナル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)~2)]．実システムでは，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，主記憶上のデータベース及び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．筆者らはデータベースマシンを用いるアプローチとして，主記憶上に常駐化されたデータベースをベクトル処理を用いて高速化する内蔵データベースプロセッサ方式と，2次記憶装置上のデータベースを高速にアクセスするフィルタリングプロセッサ方式を採用した．

内蔵データベースプロセッサ IDP[3)~5)]では，プロセッサ処理時間の短縮を目的に，従来のベクトル演算機構に加えたパイプライン方式のマージ演算機構を開発した．ソフトウェアとしては，主記憶上に読み込まれたレコード群と対応したベクトル構造を動的に作成する方式を開発し，データベース処理のベクトル化を可能にした．

また，フィルタリングプロセッサ RDSP[6)~7)]は，大型計算機のチャ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，RDSP を支援する DBMS を開発した．

本稿では，内蔵データベースプロセッサ IDP とフィルタリングプロセッサ RDSP の概要について述べる．2. では，内蔵データベースプロセッサ IDP について述べる．3. では，フィルタリングプロセッサ RDSP について示す．4. では，まとめについて述べる．

2. 内蔵データベースプロセッサ IDP の概要

2.1 IDP のアーキテクチャ

IDP は大容量主記憶を有する大型計算機を対象に開発された機構であり，大部分の情報を主記憶上に常駐化しても高速化できないような環境でのプロセッサ負荷削減が，主目的である．その基本思想は，リレーショナルデータベースの概念的な表形式のデータ構造が2次元のアレイ構造である点に着目し，ベクトル演算機構の適用によりプロセッサ処理の高速化を図ろうとするものである．

スーパコンピュータや大型機の付加機構として広く利用されてきたベクトル演算機構であるが，

† Integrated Database Processor IDP and Filtering Processor RDSP by Masashi TSUCHIDA and Shun'ichi TORII (Systems Development Laboratory, HITACHI Ltd.).

†† (株)日立製作所システム開発研究所

数値計算とは別にリレーショナルデータベースに適用するに当たっては以下の2点に大きなアーキテクチャ上の課題があった．

（1） 各ベクトルオペランド要素への指標

たとえばベクトルAとBの加算をCに格納する場合を考えると，A(i) は必ず i 番目の演算でB(i) と加算されC(i) に格納される．すなわち，マージ型の演算に現れるような各オペランド別に指標 (i, j, k) を持つことができない．IDP では，各オペランドごとに指標を持てるように拡張し，マージや探索演算のベクトル化を可能とした．

（2） ベクトル要素のデータ構造

従来はデータ型が整数や浮動小数点に制限されていただけでなく，各要素には一つの値しか格納できなかった．IDP では，デュアルベクトルと名付けた各ベクトル要素がフロント部とリア部からなる新しいデータ形式を基本としている．

従来のベクトル機構を拡張した IDP の命令形式を**図-1** に示す．デュアルベクトル命令は**表-1** に示すように，4種に分類できる．最初に開発した HITAC M-68x プロセッサグループでは最も汎用的な9命令を採用した．各演算は，リア部を比較対象としている．後継機の HITAC M-880 プロセッサグループでは効率向上のため専用命令を追加し，13 命令が実現されている．フロント部/リア部データの型や長さも M-880 ではレパートリを拡充し適用範囲の拡大を図っている．

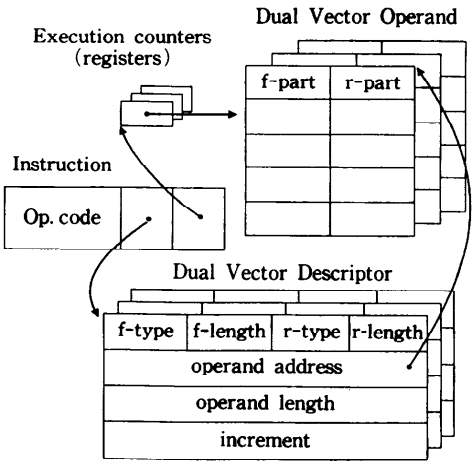

図-1 IDP の命令形式

2.2 動的ベクトル化方式

従来のリレーショナルデータベースで採用されている1レコードを単位とした処理方式は，小容量主記憶上で小数のレコードを処理する際には効率的である．しかし，2次元表構造のロウ方向のベクトルしか操作できず，この方向では各要素の型が異なり，ベクトル化が難しい．これに対して，カラム方向はデータ形式が同一でありベクトル化の可能性が高い．IDP では，大容量主記憶上での多数レコード操作を対象に，ディスク上にはレコード単位で格納されている従来のデータ形式を継承しながら，複数レコードを一括処理する方式によりベクトル化を実現した．ここでは二つの例を用いて，ベクトルを主記憶上に動的に生成する方式の具体的な手順を説明する．

図-2 は，有効なインデクス情報を使用できない処理の代表例として，二つの表の結合検索を動的ベクトル化方式で行った場合を示している．基本アルゴリズムはマージジョイン方式に基づいている．

（1） デュアルベクトル作成

最初に各テーブルについて，従来のデータ構造を動的にデュアルベクトル構造に変換する．全レコードのレコード識別子 (RID) を適当なインデクス情報から抽出し，デュアルベクトルのリア部に格納する．ここで RID は，そのレコードを格納するページ番号 (P#) とページ内の何番目かを示すスロット番号 (S#) とからなる．フロント部に，各レコードの主記憶アドレスを格納する．主記憶アドレスから，リア部にカラム値を持ったデュアルベクトルを別に作成する．フロント部には

表-1 IDP デュアルベクトル命令

命　令	機　能　概　要	応　用　例	M-68x	M-880
ソート	上昇順のマージ	集合和，ソート	1命令	5命令
結　合	マージ型同一値要素ペア取出し	集合積，結合	1命令	2命令
集合差	マージ型不一致値要素取出し	集合差	1命令	1命令
逐次探索	条件成立ペア取出し	選択，統計演算，重複排除	6命令	6命令
合　計			9命令	13命令

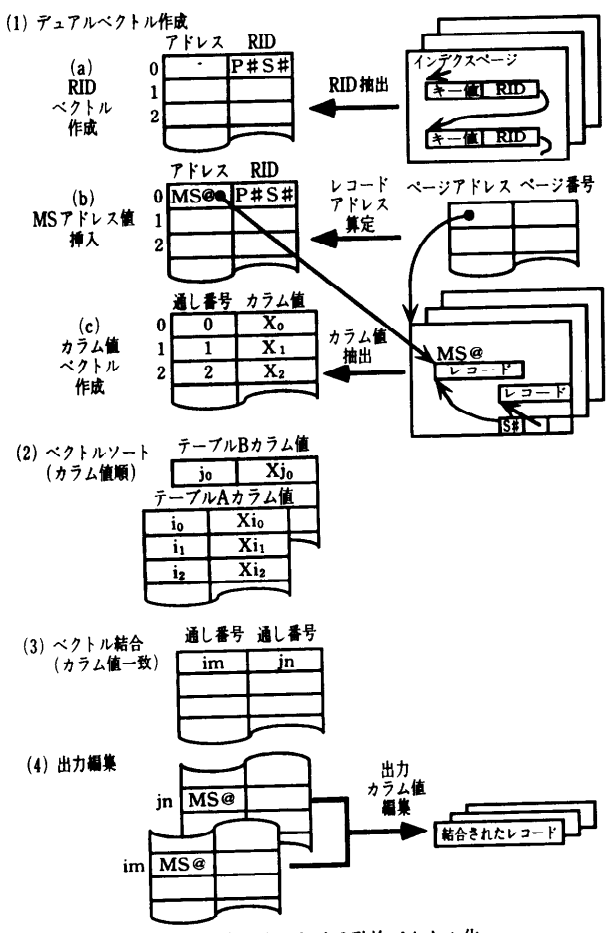

図-2 結合検索における動的ベクトル化

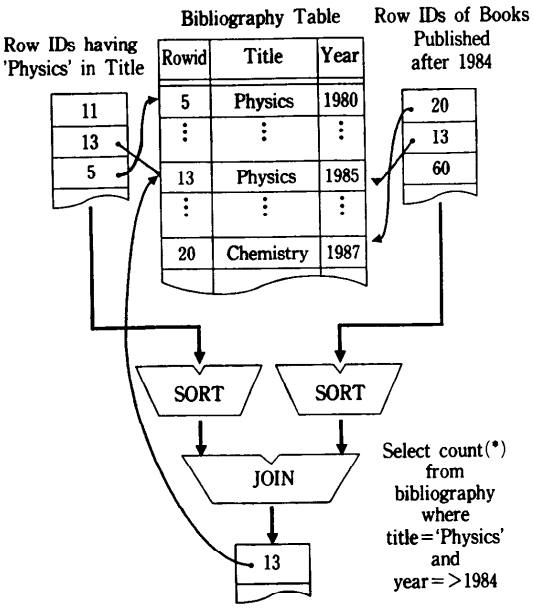

図-3 複合条件検索のベクトル化

通し番号を付ける.

(2) ベクトルソート

各テーブルについて，後者のリア部にジョインカラム値が格納されたデュアルベクトルをジョインカラム値でソートする.

(3) ベクトル結合

ソートされたデュアルベクトルをマージしながら結合する.

(4) 出力編集

フロント部に付けておいた通し番号を基に，元のレコードを参照して出力カラム値を編集する.

以上の処理において，(1)(a)の RID ベクトル作成の処理を除いて大部分の処理はベクトル化の対象とすることができる.

図-3 は，一つの表に対して複数の検索条件が指定された場合の処理手順を示している．この検索は，各条件カラムのインデクス情報を有効に使った例である．内部の処理手順は，ソートと結合の部分は図-2 と類似したものとなるがデュアルベクトルが簡単に生成できる点と，リア部しか使わない点が異なる．この場合も高いベクトル演算機構の適用率が期待できる.

2.3 性能評価

図-4 は，M-68x でのソート命令の性能を各種ソートアルゴリズムと比較したものである．横軸はソート対象ベクトルの要素数 n，縦軸は $n \log n$ で正規化した処理時間である．要素数が 1,000 を超えた場合には，通常最も高速と言われているクィックソートと比較しても数倍早いことを示している．IDP 命令が 1 万要素以上で性能が低下しているのは，キャッシュからデュアルベクトルが溢れるためである.

表-2 は，2.2 で示した二つの検索処理のプロセッサ処理時間を比較したものである．おのおのが 6,000 件の二表の結合処理では動的なベクトル化だけでも 1 桁近い高速化が実現され，IDP ハードウェアによりさらに 4 倍程度の高速化が可能である．次は，二つの AND 条件で内部的には 8,000 件が操作対象となる複合条件検索の結果である．複合条件検索の場合には比較の対象とした RDBMS に動的ベクトル化方式に相当する機能がすでにあり，ソートのアルゴリズムにはより効率

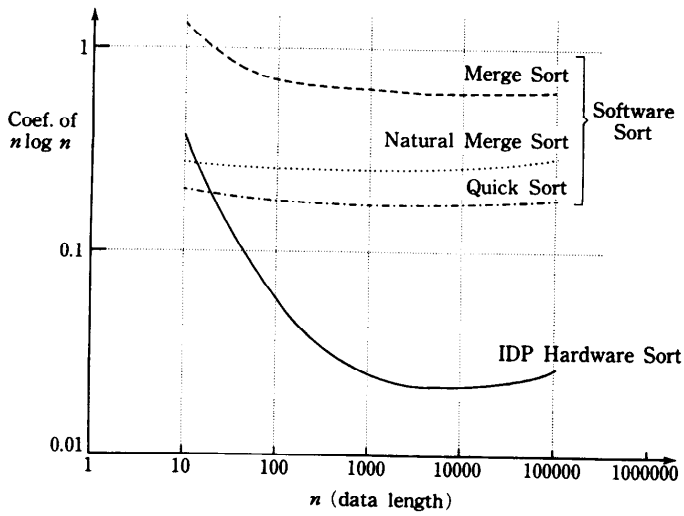

図-4　M68x の IDP ソート命令性能

表-2　動的ベクトル化方式と IDP ハードウェアの効果

検索の種類	二表の結合検索	複合条件検索
従来処理方式	4.62秒	0.072秒
動的ベクトル化方式 (IDP 無)	0.49秒	0.207秒
動的ベクトル化方式 (IDP 有)	0.11秒	0.018秒

測定マシン：M-680 プロセッサ処理時間

の良いクィックソートが採用されていたため，IDP ハードウェアがない場合には逆に遅くなっている．しかし IDP ハードウェアの適用により全体では 4 倍程度の高速化が実現できた．

3. フィルタリングプロセッサ RDSP の概要

3.1 RDSP のアーキテクチャ

図-5 は，RDSP のシステム構成である．RDSP は，チャ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，数などのデータベース演算を実行する．これによって，プロセッサ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．RDSP を実システムに適用するに当たって，以下の 2 点の課題があった．

（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行に実行される環境において，RDSP 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．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，すなわち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，RDSP はページロックの早期解放方法を採用した．

（2）　格納構造への対応

多くのフィルタリングプロセッサは，特定の格納構造にだけ対応していた．大型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，RDSP の接続インタフェースは，複数の格納構造に対応できなければならない．

RDSP は，HITAC M-640/660 プロセッサのチャネルと入出力制御装置との間でデータの転送を制御する専用のハードウェアであり，7 種のコマンドを採用している．RDSP の特徴を示す．

（a）　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パイプライン動作させる．

（b）　システム構成の拡張性

RDSP は，チャネルと入出力インタフェースで接続されるので，各種プロセッサ，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，RDSP の増設で対処し，また並列動作も可能とする．

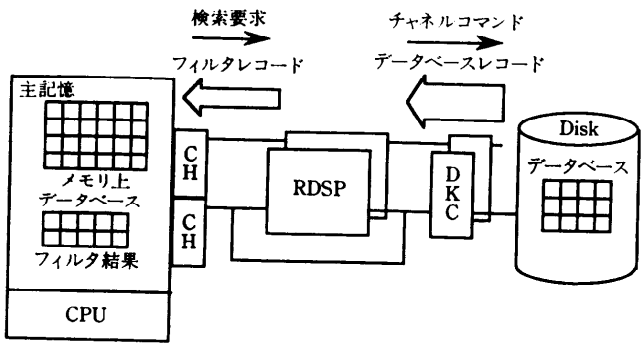

図-5　システム構成

3.2 接続インタフェース

図-6 は，チャネルと RDSP 間のインタフェースである．このインタフェースは，チャネルコマンドを拡張し，設定し

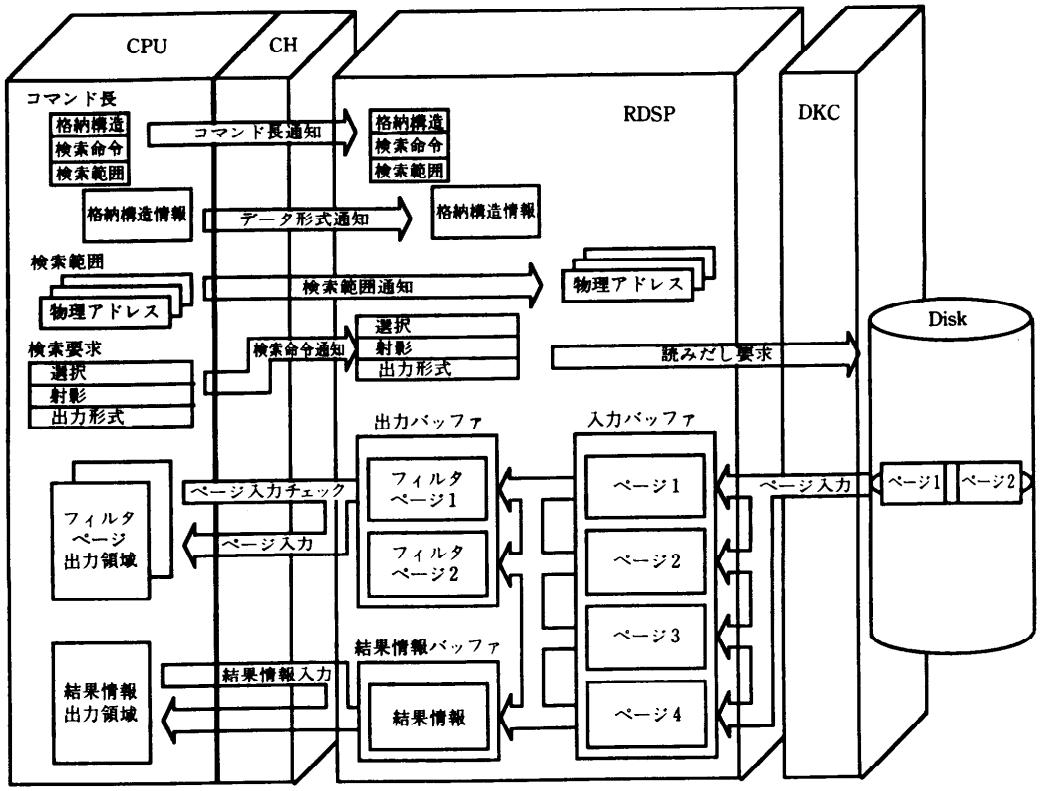

図-6 RDSP インタフェース

表-3 RDSP コマンド

コマンド種別		機能概要
制御コマンド	コマンド長通知	チャネルコマンド長の通知
	データ形式通知	データベース格納構造の通知
	検索範囲通知	ディスク上の格納範囲の通知
	検索命令通知	条件式，射影カラム，出力形式の通知
	ページ入力チェック	RDSP の転送状態の確認
データ転送コマンド	ページ入力	ページの読み込み
	結果情報入力	結果情報の読み込み

た．表-3 は，各 RDSP コマンドの機能概要である．これらコマンドの動作概要を以下に示す．

（1） コマンド設定

チャネルコマンド長，データベースの格納構造，読み込むべきディスク上の格納範囲を設定する．この構造情報で，さまざまなレコード形式を処理可能とする．

（2） 検索処理実行

検索命令通知コマンドは，条件式，射影カラム，出力形式からなる．RDSP は，コマンドを解析して，一連のチャネルコマンドを入出力装置に発行する．検索範囲通知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，RDSP はすでに読み込んであったページを検索して，出力バッファに出力形式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，条件式を満足するレコードを含まないページ，及びレコードがオーバフローしており条件式の評価ができないページなどの情報を蓄積する．

（3） 検索結果入力

RDSP が検索結果を転送可能か否かをチェックし，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．最後に，検索命令通知コマンドの実行によって蓄積された結果情報を読み込む．

3.3 ソフトウェア方式

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．表のレコードは，ページに分割，

格納される．さらにレコードは，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，レコードが処理単位である．すなわち，各アプリケーションプログラムは，一時に1レコードの単位で処理を行う．ディスク入出力もランダムにレコードが格納されるページ単位で発行される．したがって，レコードが処理の単位であった．

RDSP システムでは，データ操作が一時に複数レコードを処理可能とし，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している．RDSP へのアクセス要求は，あらかじ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でアクセスされる複数ページの単位で行われる．RDSP へのアクセス要求の終了後，条件式を満足しないレコードだけを含むページに対するロックは，RDSP の結果情報を基にして解放する．この情報を用いて，ロック占有時間を最小化する．

また，RDSP をアクセスするチャネルコマンドを生成する専用の入出力プログラムも開発した．図-6 に示すコマンド群を生成するために用いる．アクセス法は，4つのパラメタを解釈し，RDSP をアクセスする，次のパラメタからなる．

（1） 検索要求

このパラメタは，条件式，射影カラム，出力形式を指定する．検索要求は，検索命令通知コマンドに変換される．

（2） 検索範囲

入出力装置は，VSAM (Virtual Storage Access Method) 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，RDSP 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） フィルタ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．フィルタページアドレスリストから，一連の RDSP コマンドが作成される．

（4） 検索情報アドレス

これは，オーバフローレコード情報とアンロックページ情報とからなる．

3.4 性能評価

RDSP システムでは，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

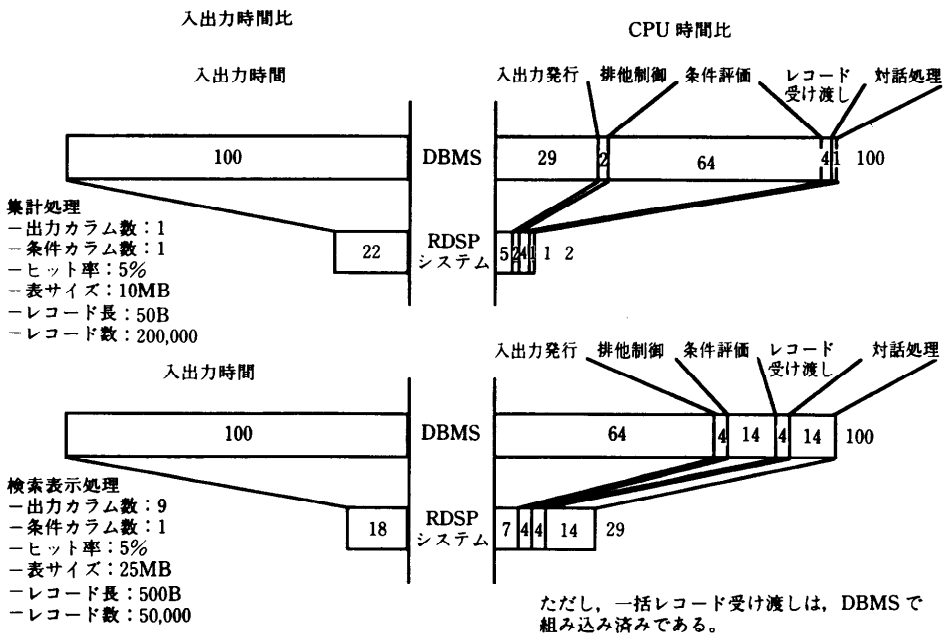

図-7 RDSP システムの削減効果

理するため，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．複数ページを一括して処理することによる入出力時間の削減効果と RDSP システム適用による CPU 時間の削減効果に起因する．すなわち，従来システムとの比較で CPU 時間が 1/5〜1/3，及び入出力時間が 1/5 に削減できる併用効果がある．

4. ま と め

本稿では，内蔵データベースプロセッサ IDP とフィルタリングプロセッサ RDSP の概要について述べてきた．IDP は M-68x に続いて，M-880 でも採用された．最も性能向上が期待できるソート/結合のマージ演算を中心にインデクス条件の有効利用の場合を含め，適用分野の拡大を図る．また，RDSP は，今後増加するデータベース量に対応した拡張が必要になる．さらに，オンライン処理と対話処理が混在するシステムへの適用拡大も図る．

参 考 文 献

1) Ozkarahan, E.: Database Machines and Database Management, Prentice-Hall, pp. 224-321 (1987).
2) 喜連川，伏見：データベースマシン，情報処理，Vol. 28, No. 1, pp. 56-67 (1987).
3) 小島，鳥居，吉住：ベクトル型データベースプロセッサ IDP，情報処理学会論文誌，Vol. 31, No. 1, pp. 163-173 (1990).
4) Torii, S., Kojima, K., Sakata, M., Yoshizumi, S. and Takahashi, M.: Accelarating Non-Numerical Processing by an Extended Vector Processor, Proc. of 4th ICDE, pp. 194-201 (1988).
5) Kojima, K., Torii, S. and Yoshizumi, S.: IDP—A Main Storage Based Vector Database Processor, Database Machine and Knowledge Base Machine, Kluwer Academic, pp. 237-250 (1988).
6) 北嶋，大曽根，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 (1989).
7) 土田，河村，中野，武藤，北嶋，米田：高速フィルタリングプロセッサの方式と性能評価，情報処理学会研究報告，Vol. 91, No. 86, pp. 25-32 (1991).

(平成 4 年 7 月 13 日受付)

土田 正士 (正会員)

1958 年生．1981 年筑波大学第三学群情報学類卒業．1983 年同大学理工学研究科修士課程修了．同年(株)日立製作所システム開発研究所入所．以来，データベースシステムの研究に従事．現在同社システム開発研究所第 6 部研究員．ACM 会員．

鳥居 俊一 (正会員)

1949 年生．1971 年東京大学工学部計数工学科卒業．1973 年同大学院修士課程修了．同年(株)日立製作所中央研究所入所．以来，大型計算機，スーパコンピュータおよびデータベース高速化の研究に従事．現在同社システム開発研究所第 6 部主任研究員．