計算機アーキテクチャ　90－2
（１９９１．１０．７）

リレーショナル・データベースマシンＲＩＮＤＡにおける サーチ処理方式

速水　治夫,　　佐藤　哲司,　　黒岩　淳一,　　中村　敏夫
ＮＴＴ情報通信網研究所

リレーショナルデータベースの非定型の検索処理ではインデックスの利用が困難なため，ソフトウェアのみで実現するシステムでは，表中の全ての行を読み出して検索条件を判定するサーチ処理の負荷が高く，多大の処理時間がかかっていた．ＲＩＮＤＡではサーチ処理を内容検索プロセッサ（ＣＳＰ）で超高速に実行している．

本論文ではＲＩＮＤＡにおけるサーチ処理方式を述べ，性能の実測評価による本方式の有効性を示す．本方式による性能向上効果は極めて大きく，従来のソフトウェアのみによる場合と比較して非定型の検索処理を最大100倍以上に高速化できた．主な提案方式は以下のとおりである．

(1)　検索条件判定をＣＳＰとソフトウェアが分担実行する本システムにおいては，検索条件の論理式の内部処理形式として乗法標準形が適している．

(2)　乗法標準形の論理式のNOTを述語の中に組み込むことにより，ＳＱＬ仕様に基づく３値論理の検索条件判定を２値論理に縮退する．ＣＳＰの高速・小規模な乗法標準形の論理式判定回路を示す．

(3)　汎用ディスク装置に対する高速サーチ処理を実現するため，データ読み出しと検索条件判定をパイプライン処理する．更に，複数ＣＳＰにより並列処理する．

Search Operation Methods in RINDA - a Relational Database Machine

Haruo HAYAMI, Tetsuji SATOH, Junichi KUROIWA, and Tosio NAKAMURA
NTT Network Information Systems Laboratories
1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 Japan

Fast search operation methods implemented in a relational database machine, RINDA, are described. RINDA was developed in order to accelerate non-indexed queries of relational database. RINDA is composed of CSPs and ROPs; the former searches rows stored in disk storage and the latter sorts rows stored in main memory. CSP tranfers pages successively to data buffers from disk by means of multi-track-read method, and searches the page in buffer before the next page coming. Therefore, the CSP can perform searching within the page transfer time, similarly to on-the-fly.

1．まえがき

リレーショナル・データベース（ＲＤＢ）はデータベースの論理構造が単純な表形式であり，その表（リレーション）の任意の行（タプル）と列（アトリビュート）の集合に対してデータ操作を行うことができる．そのためのデータ操作は，ＩＳＯおよびＪＩＳで標準化されたデータベース言語ＳＱＬ[1)・2)]に見られるように高水準な言語インタフェースとなっている．

また，ＲＤＢが提案されて以来，ＲＤＢ処理を高速化するデータベースマシンの開発が盛んに行われている[3)・4)]．これは，ＲＤＢのデータ操作の水準が高く，ソフトウェアのみの実現では高速化に限度があり，近年の半導体を始めとするハードウェアテクノロジーの進歩を背景として，専用ハードウェアの支援による高速化が有効となってきたことによる．

これらのアーキテクチャの中に，ディスク装置と汎用計算機との間のＩ／Ｏボトルネックの解消を目的としたフィルタプロセッサを使用する方式も提案されている[5)・6)]．著者らもフィルタプロセッサ型の内容検索プロセッサ（ＣＳＰ:Content Search Processor）を構成要素の一つとする商用のリレーショナル・データベースマシンＲＩＮＤＡ[7)・8)]を開発した．

ＲＤＢの非定型の検索処理ではインデックスの利用が困難なため，ソフトウェアのみで実現するシステムでは，表中の全ての行を読み出して検索条件を判定する処理（以下，サーチ処理と呼ぶ）の負荷が高く，多大の処理時間がかかっていた．ＣＳＰは，サーチ処理を高速化することを目的とし，汎用計算機ＤＩＰＳシリーズ[9)・10)]と，データベースが格納されている汎用ディスク装置との間に入出力インタフェースで接続され，ＤＩＰＳ上のソフトウェアであるデータベース管理システム（ＤＢＭＳ）によって制御される．以下では，ＲＩＮＤＡとＤＩＰＳおよびこれらを制御するＤＢＭＳをＲＩＮＤＡシステムと呼ぶ．

本論文では，ＲＩＮＤＡシステムのサーチ処理におけるＳＱＬ検索条件の判定方式，複数ＣＳＰによる並列サーチ処理方式およびＣＳＰの実現方式について述べる．

以下，２章でＲＩＮＤＡ全体の概要を，３章でサーチ処理における検索条件判定方式を，４章で複数ＣＳＰによる並列サーチ処理方式を，５章でＣＳＰ自体の実現方式を述べ，６章で100万件のデータベースを用いた性能評価結果を示す．

2．ＲＩＮＤＡの概要

ＲＤＢの主要な処理は，定型の検索処理・更新処理と非定型の検索処理である．定型の検索処理の代表的な例は表の中からユニークなキーを用いて単一の行を選択する処理であり，インデックスを使用することにより，ソフトウェアのみでも十分に実用的な性能が実現されている．

他方，非定型の検索処理には，あらかじめインデックスの作成されていない列に対して条件を指定した検索，インデックスの作成が困難な文字列データの列に対する任意文字列の部分一致検索，およびインデックスの作成されていない列どうしで複数の表を結合する処理などがある．これらを従来のソフトウェアのみのシステムで実行すると多大の処理時間がかかっていた．その原因は，インデックスが使用できないため，表中の全ての行を読み出して検索条件の判定を行う処理（サーチ処理）や，結合処理のため表を特定の列の値の順序で並び換えるソート処理の負荷が大きく，多大の処理時間がかかっていたことによる[11)]．

上記の問題を解決するために開発されたＲＩＮＤＡでは，サーチ処理とソート処理をそれぞれＣＳＰと関係演算プロセッサ（ＲＯＰ:Relational Operation Accelerating Processor）[12)]とで超高速に実行する．ＣＳＰ，ＲＯＰはそれぞれ独立の入出力インタフェースでホスト計算機（ＤＩＰＳ）と接続され，ＤＩＰＳ上のソフトウェアであるデータベース管理システム（ＤＢＭＳ）によって制御される．ＣＳＰとディスク制御装置（ＤＫＣ）も入出力インタフェースで接続される．ＣＳＰは，ディスク装置（ＤＫ）に格納された表を読み出し，検索条件に合致する行の必要な列を抽出してホスト計算機に転送する．ＲＯＰはホスト計算機から転送さてくる表をソートし，その結果をホスト計算機に送り返す．ＣＳＰ，ＲＯＰが実現している主な機能をそれぞれ表１，表２に示す．ＲＩＮＤＡが実現する機能は非定型の検索処理のみであり，その他の機能は従来どおりＤＢＭＳによって実現される．業務処理プログラムとのインタフェースはＳＱＬに準拠しており，業務処理プログラムからＲＩＮＤＡを意識する必要はない．

3．検索条件判定方式

3.1　分割実行方式

ＳＱＬの検索条件は，述語と呼ばれる真偽判定の最小単位の条件式をブール演算子AND，OR，NOTで結んだ論理式で与えられる．ＣＳＰはＳＱＬの検索機能を基本的にそのまま実行することにより，ホスト計算機の負荷を極力オフロードする．しかし，使用頻度が低い一部の機能はＣＳＰで実現しておらず，ホスト計算機上のＤＢＭＳが実行する．業務処理プログラムから指定されたＳＱＬ文の中にＣＳＰが実行できない述語が含まれる場合は，ＳＱＬ文全てをホスト計算機上のＤＢＭＳが実行するのではなく，ＣＳＰが実行できるところまで実行して絞り込んだ結果に対してＤＢＭＳが残りの機能を実行する．この方式に適した検索条件の論理式の形式は乗法標準形（論理変数（ここでは述語）を論理和で結合する項（論理和項）を論理積で結合する形式）である．ＤＢＭＳは業務処理プログラムから与えられたＳＱＬ文の論理式を乗法標準形に変換し，ＣＳＰで実行可能な述語のみからなる論理和項を論理積で結合した部分をＣＳＰへ転送する．ＣＳＰが処理した結果に対して，ＤＢＭＳは残りの論理和項を実行する．他方，論理式を加法標準形（述語を論理積で結合する項（論理積項）を論理和で結合する形式）に変換し，各論理積項を分割して処理すると，処理結果を併合しなければならない．

3.2　2値論理演算への縮退方式

ＳＱＬではデータ項目の値が存在しない時は，ゼロやスペースなどの本来の意味と異なる値でなくNULLと表現する．これに伴い述語の判定結果は真，偽以外に述語の中のデータにNULLが含まれた場合の不定の3値の真偽値となり，AND，OR，NOTは図1の真理値表で定義された演算子となる．

ＣＳＰの実現において，3値の論理演算をその定義どうりに実行するより，2値の論理演算に縮退できれば，その回路構成が単純となる．

検索条件の判定結果は，上記の3値に対して定義された論理演算の結果の3値である．しかし，検索条件の判定結果により行を選択するときには，検索条件の判定結果が真の行のみが選択される．すなわち，最終的には不定は偽に置換される．

図1の真理値表で特に注目すべきは，NOT(不定)はやはり不定となることである．即ち，行の選択においてNOT(不定)は偽と等価であるのに，不定を単純に偽に置き換えてしまうと，NOT(不定)⇨NOT(偽)＝真となり不合理を生ずる．

従って，検索条件の論理式にNOTがないことが，述語判定結果の不定を偽に置換し，検索条件判定における3値の論理演算を2値に縮退可能な条件である[13]．

任意の検索条件を上記検索条件に変換する方法は，検索条件の論理式をNOTが1つの述語だけにかかる標準形に変換し，その結果NOTがついた述語があれば，NOTを述語中に組み込むことである．

ＲＩＮＤＡシステムでは，前節に述べたようにＤＢＭＳが論理式を乗法標準形に変換する．この変換後，上記標準形の中に，NOT(述語)があれば，NOTを述語中に組み込む．即ち，比較述語の場合は比較演算子を逆にし，他種の述語の場合は述語中に入れる．例えば，(NOT(列1 > 10))は (列1 <= 10)に，(NOT(列1 LIKE '%A%'))は(列1 NOT LIKE '%A%'))に変換する．

この結果，ＣＳＰは述語判定において不定を偽に置換し，2値の論理演算で検索条件の判定が可能となる．

4．並列サーチ処理方式

サーチ処理は表全体を読み出すため，表のデータ量に比例した処理時間を要する．表を複数のＤＫ（ディスクボリューム）に分散格納し，複数（ｎ台）のＣＳＰで並列サーチ処理すれば，処理時間を更に短縮することができる．ＲＩＮＤＡシステムでは，格納時，検索時とも，業務処理プログラムからは複数ＤＫを使用していることを意識させることなく，サーチ処理時間を１／ｎに短縮している．

4.1　分散格納

一般に複数台のＤＫに格納されているデータの並列サーチ処理の効率を上げるために考慮すべき事項は以下である．

①　各ＤＫのデータ量が均等であること．

②　検索時に各ＤＫから選択されるデータ量が均等であること．

ＲＩＮＤＡシステムでは検索条件が予想できない非定型検索を対象とするものであるから，格納行を特定の列の値または複数列の連結値によって，上記②の条件を満足するように複数ＤＫに振り分けて格納することは不可能である．このため上記①の条件を満足するように，行は格納要求順にページに格納し，ページ単位で循環的に各ＤＫに格納する．

4.2　並列サーチ処理

ＤＢＭＳが検索対象のページとＤＫの対応関係から独立動作可能なＤＫを選択し，そのＤＫをサーチ処理

可能なＣＳＰに対して一斉にサーチ処理開始を指示し，複数ＣＳＰによる並行処理を行う．ＣＳＰごとの処理結果はＤＢＭＳがとりまとめて業務処理プログラムに返却する．

5．ＣＳＰの構成

5.1　ページ・サーチ処理方式

サーチ処理の実行には以下の二つの方法がある．

(1)　ストリーム処理方式：ディスクの回転に同期して，読み出されるデータの流れに沿って検索条件の判定を実行する．

(2)　ページ・サーチ処理方式：ディスク装置からの読み出しデータをページ単位でバッファに格納した後，検索条件の判定を実行する．

ＣＳＰでは以下の理由でページ・サーチ処理方式を採用した．

①　汎用ディスク制御装置はページ単位で，記録データのエラー検出・訂正機能を有している．本機能はページの全データを読み出した後に有効であり，エラー訂正の為にはページがバッファメモリ上に存在している必要がある．

②　ストリーム処理方式では，ページの先頭から有効データを格納するなど本処理方式に相応しい格納構造を採る必要があるが，ページ・サーチ処理方式ではページ内の格納構造を自由に採ることが可能である．ＲＩＮＤＡシステムでは既存ＤＢＭＳ単独の処理と共存するため，データベース格納構造を変更していない．

③　古典的なストリーム処理方式が提案された当時は，メモリ素子が高価でページバッファを複数個設けることはコスト的に無理であった．現状はメモリ素子の低価格化が進んでおり，ページバッファ用メモリのコストの占める割合は大きくない．

ＣＳＰはＤＫからシリンダ単位のマルチトラックリードでデータ読み出しを行い複数の入力バッファに格納し，ページ単位でデータ読み出しと検索条件の判定をパイプライン処理する．このため，実効的にストリーム処理方式と同様にデータ転送時間内のサーチ処理（いわゆるオンザフライ処理）を実現している．

5.2　ＣＳＰのブロック構成

ＣＳＰのブロック構成を図２に示す．ＣＳＰはホスト計算機／ＤＫＣとのインタフェースを制御するチャネルインタフェース制御部／ＤＫＣインタフェース制御部，入出力ページを一時格納する複数の入／出力バッファ，述語を判定する述語判定部，検索条件の論理式を判定する論理式判定部，入力バッファ中から検索対象行の必要列を抽出して述語判定部と出力バッファへ転送する行・列抽出部および全体を制御する主制御部から構成される．

ＣＳＰは，ＤＢＭＳが作成しチャネルコマンドで転送してきたオーダ（制御情報）に従ってサーチ処理を実行する．オーダ中には，サーチ処理対象のＤＫアドレス，ページアドレス，ページ容量などの物理制御情報と，表１の検索機能を指定した論理制御情報が含まれている．ＣＳＰは受け取ったオーダの物理制御情報をもとにＤＫに対しチャネル・コマンドを発行し，サーチ処理対象のページをマルチトラックリードにより連続的に読み出す．同時に，論理制御情報に基づき検索条件の判定と出力列の抽出を実行する．この動作を間断なく行うために，ＤＫから読み出したページを格納する入力バッファと検索条件合致行の抽出列をホスト計算機に転送するための出力バッファは複数個を交代で使用する．出力バッファ中に検索結果のデータがページ単位にまとまる毎に，検索条件判定とは非同期に検索結果をホスト計算機に転送する．

検索条件判定は述語判定と論理式判定とにステージを分けて処理している．また述語判定と同時に，出力列を抽出し出力バッファに転送している．検索条件が偽の場合は実行中の（又は完了した）列抽出処理を中断・無効化し，次の行の抽出列を出力バッファの前行と同一アドレスに上書きしていく．同一行の述語判定と列抽出を同時処理することにより，入力バッファから列読み出しの重複が避けられる．

述語判定部では各述語対応に専用回路で判定する．

5.3　論理式判定部の構成

ＳＱＬ仕様に基づく検索条件の論理式判定は３値論理であるが，データにNULLの存在を許していないシステムでは不定が無いため，検索条件の論理式判定は２値論理であり，２値論理での論理式判定をハードウェアで高速に処理する方法には，①論理式を加法標準形で表現し連想メモリを使用する方法[14]，②論理式を主加法標準形で表現しビットアレイを使用する方法[15]などが提案されている．こららの方法は，論理式にブール演算子NOTが使用されていても判定可能であるが，①は専用の連想メモリを使用するか，等価なコンパレータが必要であり回路は大規模となる．②は述語数をｎとするとビットアレイのビット数は2^nであり，ｎの増加とともにビット数と検索準備時間が爆発的に増加する．

ＣＳＰに与えられる論理式はNOTの無い乗法標準形であり，これに適した判定回路を開発した．ＣＳＰの

論理式判定部のブロック図を図3に示す．検索準備で，ＤＢＭＳより転送されたビットマップ表現の論理式を論理式ビットマップメモリに格納する．検索実行では，述語判定結果がレジスタにそろった時点で，ビットマップメモリから論理和項のビットマップを１ワードずつ読み出して，述語判定結果レジスタ間のマスク論理和演算を論理和項の数だけ繰り返して実行する．マスク論理和演算が全て１のとき検索条件は真である．マスク論理和演算が１回でも０となった時点で検索条件は偽と判定できるため，マスク論理和演算の平均回数は下がる．

本論理式判定部が小規模となる理由は以下である．

①　ＣＳＰに与えられる論理式にNOT(X_i)が無いため，論理和演算にコンパレータを使用する必要がない．

②　論理和項のビットマップをメモリから読み出して，論理和演算を繰り返す方式とした．

②により論理式判定の時間は長くなるが，ＤＫからのデータ転送時間とバランスした時間内に終了する．

6．性能評価

6.1　評価条件

ＣＳＰによる非定型の検索処理の高速化効果を拡張ウィスコンシン・ベンチマーク[16),17)]により評価した．評価に使用した表の大きさは100万行であり，各行はシステム制御情報を除いて208バイト長である．評価に使用したシステムは，ホスト計算機が小型汎用機のDIPS-V30E[10)]，ＣＳＰが１台または２台，容量が1.3Gバイト，データ転送速度が3Mバイト／秒のディスク装置およびディスク制御装置各２台で構成した．

処理時間を実測した問合せを表３に示す．問合せはベンチマーク標準の問合せでサーチ処理の評価に使用できるＱ１～Ｑ３，およびＣＳＰ単独の処理時間を評価するためのＱ４～Ｑ７からなる．Ｑ４はＱ１，Ｑ２と同様な条件で検索条件合致件数が１件となるものであり，Ｑ５，Ｑ６は検索結果データを転送せずに行数だけをカウントする問合せである．Ｑ７はＣＳＰの論理式判定部の性能を評価するため，検索条件の述語数を変化させた問合せである．

処理時間は，ＤＢＭＳが評価プログラムからＳＱＬ文を受け取ってから検索結果を全て評価プログラムに返却し終わるまでの経過時間である．処理時間はＣＰＵ時間とＩ／Ｏ時間に分けて測定した．

6.2　評価結果

ＣＳＰを使用しない場合と，１台または２台使用する場合の処理時間を比較すると図４および図５のようになる．各図において処理時間比はＣＳＰを１台使用した場合の処理時間を１とした相対値である．ＣＳＰを使用した場合のＩ／Ｏ時間はほとんどＣＳＰのサーチ処理時間である．図４から，以下のことがわかる．

①　ホスト計算機のＣＰＵ時間は大幅に短縮される．

②　ＣＳＰのサーチ処理時間はＣＳＰを使用しない場合のＩ／Ｏ時間より短い．

③　ＣＳＰのサーチ処理時間はＣＳＰ台数に反比例する．

④　ＣＳＰを２台使用することによる非定型の検索処理の性能向上効果は，最大100倍以上である．

図５から，述語数が増加してもＣＳＰのサーチ処理時間は増加しないことがわかる．

6.3　考察

（１）性能向上要因

ＣＳＰにより上記の性能向上が実現できた要因をまとめると以下となる．

（ｉ）ホスト計算機のＣＰＵ時間短縮

非定型の検索処理をソフトウェアで実行した場合，サーチ処理を表中の全ての行に対して行うため，膨大なＣＰＵ時間がかかっていた．ＣＳＰの使用によりこれらの処理の大半をオフロードでき，ＣＰＵ時間を大幅に短縮した．各問合せについて要因を分析すると以下となる．

(a)　述語の種類によってＣＰＵ負荷が異なるため性能能向上効果は異なる．特に，汎用計算機では負荷の重い文字列検索（LIKE述語）を使用した問合せ（Ｑ６）の性能向上効果は著しい．

(b)　ＣＳＰはＳＱＬの全ての機能を実現しておらず，例えばMIN関数はＣＰＵで処理する．ただし，ＣＳＰがMIN関数の対象列のみからなる一時表にしてＣＰＵへ転送するため，ＣＰＵ負荷は大幅にオフロードされ，性能は向上する（Ｑ３）．

(c)　ＤＢＭＳから性能評価プログラムへ検索結果を行単位で返却する処理はオフロードされていないため，検索条件が同様の場合，処理結果行数が増加するに従って性能向上効果は相対的に小さくなる（Ｑ４→Ｑ１→Ｑ２）．

（ｉｉ）Ｉ／Ｏ時間短縮

一つの表を複数のＤＫに分割格納し，複数（ｎ台）のＣＳＰにより並列サーチ処理することによりサーチ処理時間は１／ｎになる．

また，ＣＳＰはディスク装置から１シリンダ内の全ページを途中でのディスクヘッドの機械的動作無しに連続的に読み出すために，Ｉ／Ｏ時間そのものも短縮

される．

（２）論理式判定部の構成評価

Ｑ７の問合せの検索条件はｎ個の述語をANDで結んだものであり，前半のｎ－１個の述語は常に真であるため，論理式判定部の論理和演算は常に最大のｎ回実行されている．図５に見られるように述語数によらずＣＳＰ処理時間は一定であるため，論理式判定部は論理和演算を繰り返す方式で充分であると評価できる．

（３）効果の有効範囲

文献８において１万行のウィスコンシン・ベンチマークによる性能評価結果を報告した．本論文の評価に使用した表の大きさは100万行である．両評価において対応する問合せの性能向上効果（相対値）はほぼ同一である．ＣＳＰを用いたサーチ処理時間は，表の大きさ（データ量）に比例し，汎用計算機による検索処理時間も，インデックスを使用できない場合は表の大きさに比例するため，表の大きさに依らず性能向上効果は一定となることが確認できた．

７．むすび

本論文では，ＲＩＮＤＡシステムのサーチ処理方式およびＣＳＰの実現方式を述べた．本方式による性能向上効果は極めて大きく，従来のソフトウェアのみによる場合と比較して検索処理を最大100倍以上に高速化できた．提案方式を要約すると以下のとおりである．

(1) ＲＩＮＤＡシステムではサーチ処理を基本的にＣＳＰが実行する．しかし，一部検索機能をＤＢＭＳが分担して実行するため，ＤＢＭＳは与えられた検索条件の論理式を乗法標準形に変換し，ＣＳＰが実行可能な部分を転送する．これによりＣＳＰがサーチ処理した結果をＤＢＭＳが引き継いで処理する．

(2) ＤＢＭＳは上記論理式の変換過程において，論理式にNOTを含まない乗法標準形に変換してＣＳＰへ転送する．これにより，ＳＱＬ仕様に基づく３値論理の検索条件判定を等価的に２値論理に縮退可能となり，ＣＳＰは，述語判定結果の論理式判定に小規模なマスク論理和回路で高速に実行する．

(3) ＲＩＮＤＡシステムでは一つの表を複数のＤＫに分割格納し，複数（ｎ台）のＣＳＰで並列サーチ処理し，サーチ処理時間を１／ｎに短縮している．

(4) ＣＳＰはシリンダ単位のマルチトラックリードでページの連続読み出しを行い，ページ単位でデータ読み出しと検索条件の判定をパイプライン処理する．

ＲＩＮＤＡは現在いくつかのユーザシステムに導入されており，今後は実システムにおいて，ＣＳＰのサーチ処理時間などを評価する予定である．

謝辞 ＲＩＮＤＡの開発を推進していただいたＮＴＴ研究開発技術本部の戸田巌本部長，ＮＴＴ情報通信網研究所の石野福彌所長，松永俊雄主席研究員，拝原正人情報処理研究部長を始め関係者皆様に深謝いたします．

文　献

(1) ISO 9075:"Information processing systems－Database language SQL" (1987).
(2) JIS X 3005:"データベース言語　ＳＱＬ"(1987).
(3) 喜連川優，伏見信也:"データベースマシン",情報処理，Vol.28,No.1,pp.56-67 (1987).
(4) 清水　康:"データベースマシンの動向",アドバンスト・データベース・システムシンポジューム論文集，pp.31-40 (1987).
(5) Babb E.:"Implementing a Relational Database by Means of Specialized Hardware",ACM Trans. Database Syst., vol.4, no.1, pp.1-29(1979).
(6) Ozkarahan,E.A. and Penaloza,M.A.:"On-the-Fly and Background Data Filtering System for Database Architectures", New Generation Conputing 5, OHMSHA and Springer-Verlag (1987).
(7) 速水治夫，井上　潮，福岡秀樹，鈴木健司:"リレーショナルデータベースプロセッサＲＩＮＤＡのアーキテクチャ",情処学計算機アーキテクチャ研資，88-A　RC-73-12,pp.85-92 (1988).
(8) 井上　潮，速水治夫，福岡秀樹，鈴木健司，松永俊雄:"データベースプロセッサＲＩＮＤＡの設計と実現",情処学論，Vol.31,No.3,pp.373-380 (1990).
(9) 小柳津育郎，塩川鎮雄，木ノ内康夫，安保　進:"DIPS-11/5E　シリーズの実用化",ＮＴＴ研実報，Vol.36,No.1,pp.49-56 (1986).
(10) 矢沢良一，平野正則，山口利和，岡田靖史:"DIPS-V30E のハードウェア構成",ＮＴＴ研実報，Vol.37,No.9,pp.523-532 (1986).
(11) 井上　潮，北村　正，速水治夫，中村敏夫:"情報提供サービスに適用可能な超大規模リレーショナル・データベースマシン",情処学データベースシステム研資，85-DB-47-5 (1985).
(12) 武田英昭，佐藤哲司，中村敏夫，速水治夫:"関係演算高速化プロセッサ",情処学論，Vol.31,No.8,pp.1230-1241 (1990).
(13) 速水治夫:"ＳＱＬ探索条件判定におけるＮＵＬＬ処理の簡単化",情処学論，Vol.32,No.1,pp.71-76 (1991).
(14) Paul A.Beaven :"INTERACTIVE DATA RETRIEVAL APPARATUS",United States Patent 4433392 (1984).
(15) 山田八郎，永井　肇，高橋恒介:"内容検索型バブルメモリの性能検討",信学技報，EC83-44,pp.11-pp.20 (1983).
(16) Bitton,D.,DeWitt,D.J. and Turbyfill,C.:" Benchmarking Database Systems - A Systematic Approach",CSTR #526,Univ. of Wisconsin - Medison (1983).
(17) DeWitt,D.J. et al.:"A Single User Evaluation of the GAMMAR Database Machine",Database Machines and Knowledge Base Machines,pp.370-386,Kluwer Acandemic (1988).

表1　CSPの主要機能

機能		説明
述語判定	比較述語	<列><比較演算子><定数> の条件判定.
	IN述語	<列> [NOT] IN <定数リスト> の条件判定.
	LIKE述語	<列> [NOT] LIKE <文字列> の条件判定.
	NULL述語	<列> IS [NOT] NULL の条件判定.
探索条件判定		述語の AND/OR による任意の論理式判定.
出力列の抽出		上記探索条件を満たす行から任意の列の出力.
集合関数演算		上記探索条件を満たす行数の計数 (COUNT(*)).

表2　ROPの主要機能

機能	説明
ソート	指定された列の値（ソートキーと呼ぶ）による昇順または降順への行の並び換え. ソートキーは任意の順序の複数列により構成可能
ふるい落とし	ソートキーの値による2つの表から結合可能性のない行の除去. 片方又は両方の表から除去が可能
重複計数／除去	ソートキーの値が重複する行数の計数, および2番目以降の行の除去.

表3　評価に使用した問合せ

問合せ		対応するSQL文
Q1	選択(1%)	SELECT * FROM thuk WHERE unique2>=500000 AND unique2<510000
Q2	選択(10%)	SELECT * FROM thuk WHERE unique2>=500000 AND unique2<600000
Q3	スカラMIN	SELECT MIN(unique2) FROM thuk
Q4	選択(1件)	SELECT * FROM thuk WHERE unique2=999999
Q5	スカラ計数	SELECT COUNT(*) FROM thuk WHERE unique2>=500000 AND unique2<510000
Q6	文字列検索	SELECT COUNT(*) FROM thuk WHERE stringu2 LIKE '%ABC%'
Q7	複数述語（論理式判定部評価）	SELECT COUNT(*) FROM thuk WHERE unique2<1000000 AND ・・・・・・・・・・・・・・・ unique2<1000000 AND unique2<10000 （n-1)個

AND	真	偽	不定
真	真	偽	不定
偽	偽	偽	偽
不定	不定	偽	不定

OR	真	偽	不定
真	真	真	真
偽	真	偽	不定
不定	真	不定	不定

NOT	真	偽	不定
	偽	真	不定

図1　真理値表（3値論理）

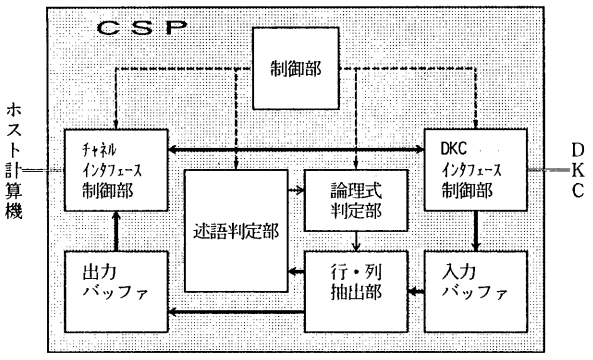

図2　CSPのハードウェア構成モデル

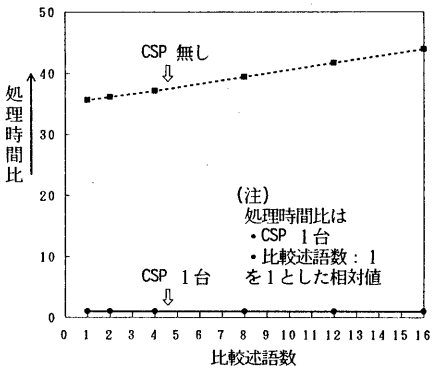

図5　CSP論理式判定部の性能評価

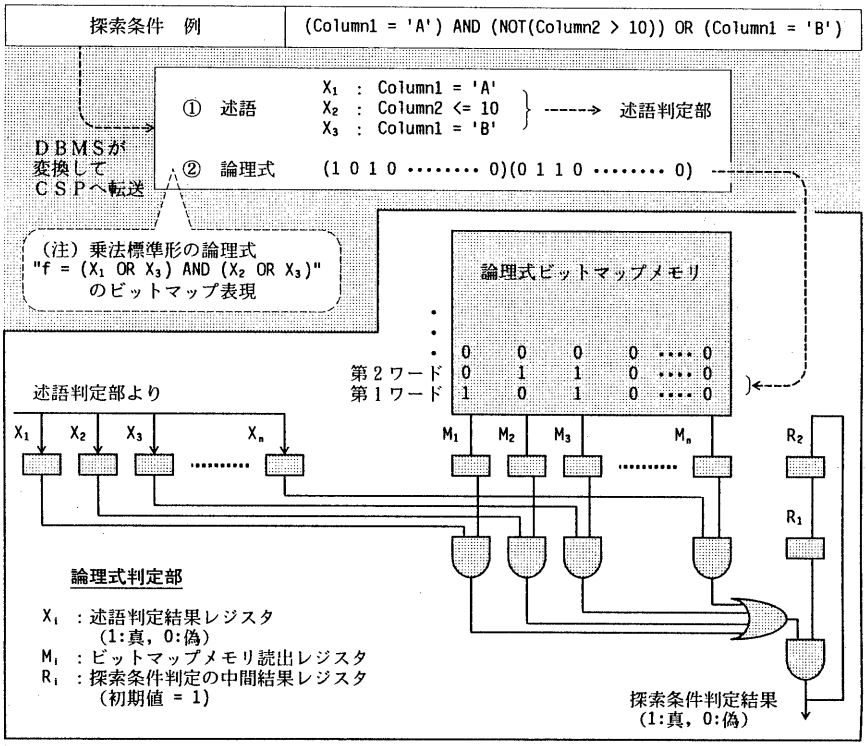

図3　論理式判定部の構成

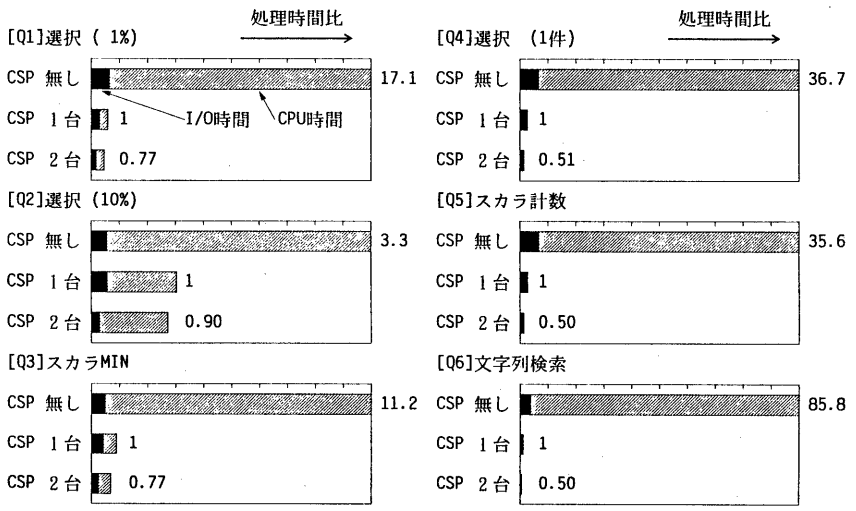

図4　ＣＳＰによる性能向上効果