

## データベースプロセッサ RINDA における 問合せ処理のアクセスパス決定方式†

芳 西 崇<sup>††</sup> 板 倉 一 郎<sup>††</sup>  
中 村 敏 夫<sup>††</sup> 井 上 潮<sup>††</sup>

関係データベースプロセッサ RINDA による処理と、従来のソフトウェアによる処理の両方が使用可能なデータベース管理システム (DBMS) における、問い合わせの最適なアクセスパス決定方式について述べる。RINDA は、関係データベースに対する索引の利用が困難な非定型の選択、および CPU 負荷の重いソート、結合等を含む検索処理の高速化を目的に開発されたデータベースプロセッサであり、汎用計算機上の DBMS により制御される。この DBMS は、RINDA による検索処理 (RINDA 処理) と、従来のソフトウェアによる検索・更新処理 (ソフト処理) の自動的なアクセスパス選択機能を有しており、ユーザに RINDA の使用有無を意識させることなく、最適なアクセスパスによる問合せの実行を可能としている。アクセスパス選択は、両方に共通な最適化部において、機能判定、性能判定の順で行われる。性能判定は、両処理をモデル化して得た I/O 回数を元に、明らかにソフトウェアの方が高速と推定される場合にのみ、ソフトウェア処理とする方式である。また、その判定基準は、SQL 構文情報、および索引定義情報のみから導いている。本論文では、RINDA 処理とソフト処理の両方を使用可能とする DBMS と最適化機構の概略を述べた後、両処理のモデル化から、最適なアクセスパス判定基準の導出までを示す。

### 1. はじめに

RINDA (Relational Database Processor)<sup>1)~3)</sup> は、関係データベースに対する索引の利用が困難な非定型の検索処理を高速化することを目的として開発されたデータベースプロセッサである。RINDA は、汎用計算機にとって負荷の重いサーチ、ソート等の処理を専用ハードウェアで超高速に実行することにより、従来の汎用計算機上のソフトウェアによるデータベース処理と比べて最高 100 倍以上の高速処理を可能とした<sup>4)</sup>。

RINDA は、データベース処理全体のうち非定型の検索機能をサポートしており、定型の検索・更新機能は具備していない。そのため、汎用計算機上のデータベース管理システム (DBMS) が RINDA の制御を行うアーキテクチャを採用している。筆者らは従来の DBMS に RINDA 制御部を付加し、ソフトウェアによる処理 (ソフト処理) と RINDA による処理 (RINDA 処理) を統合した DBMS の開発を行ってきた<sup>5)</sup>。その結果、同一のデータベース (DB) に対して、① 1 人のユーザが両処理を混在させた実行、② 両処理が混在した複数ユーザの同時実行、の両方を実現

し、ユーザは、RINDA 処理による非定型処理と、ソフト処理による定型処理を、共通インタフェースで利用可能となった。

両処理を統合した DBMS では、ユーザは RINDA の使用有無を意識せずに SQL<sup>6)</sup> 文を記述できることが望ましい。先に述べたように、RINDA は機能上、非定型検索処理のみをサポートしている。また性能上、ソフト処理で索引が有効に利用できる場合、例えばユニークなキーを利用した 1 行検索では、ソフト処理の方が高速になる。したがって、ユーザに RINDA の使用有無を意識させず、かつ高速な DB 処理を提供するためには、機能上、性能上から両処理の最適なアクセスパス選択が重要となる。特に性能上のアクセスパス選択は、全く異なる 2 つの処理コストを比較しなければならない難しさがある。従来、ソフト処理の最適化方式は数多く提案されてきているが<sup>7), 8)</sup>、ハードウェアとソフトウェアの DB 処理を統合した DBMS の最適化方式の研究はされていない。

本論文では、RINDA 処理とソフト処理が統合された DBMS の最適なアクセスパス決定方法について述べる。第 2 章では、RINDA システムのハードウェア構成と、両処理を統合した DBMS 構成を述べる。第 3 章では、アクセスパス決定方式の概要を述べ、第 4 章では、評価モデルを示す。第 5 章では、評価モデルに基づくアクセスパスの判定基準を決定する。

† Access Path Selection of Queries Using RINDA—A Relational Database Processor by TAKASHI HONISHI, ICHIRO ITAKURA, TOSHIO NAKAMURA and USHIO INOUE (NTT Network Information Systems Labs.).

†† NTT 情報通信網研究所

## 2. RINDA システムの概要

### 2.1 ハードウェア構成

汎用計算機に RINDA を接続したシステム構成例を **図 1** に示す。RINDA は汎用計算機にとって負荷の重いサーチ処理とソート処理を高速化するための専用ハードウェアであり、内容検索プロセッサ (CSP: Content Search Processor)<sup>9)</sup> と関係演算プロセッサ (ROP: Relational Operation Accelerating Processor)<sup>10)</sup> から構成されている。CSP は、ディスクに格納された表をサーチし、条件に一致した行を選択し、結果を汎用計算機へ転送する。ROP は、汎用計算機から転送された表をソートし、結果を汎用計算機へ返却する。また結合処理時にはソートに加え、結合対象外の行をふるい落とす機能も有している。CSP, ROP は独立の入出力インタフェースで汎用計算機に接続されており、並列動作が可能である。

### 2.2 DBMS 構成

RINDA 処理とソフト処理は、処理方式が根本的に異なる。例えば、ソフト処理が 1 行単位で処理するのに対し、RINDA 処理は複数行を一括処理し、中間結果を一時表の形で引き継ぎながら処理する。したがって DBMS 構成は、基本的に RINDA 処理とソフト処理は独立とし、データベース処理として集中制御するために必要な機能の一部を共通化する構成とした (**図 2**)<sup>9)</sup>。共通化した機能の概要を以下に示す。

#### ① 言語解析機能

ユーザインタフェースを同一にするため、SQL 文の構文解析、意味解析等を共通的に処理する。

#### ② 共通最適化

3 章で示すように最適化は共通最適化と個別最適化の 2 段階で行う方式とした。共通最適化では、機能上および性能上のアクセスパス判定を共通的に行う。

#### ③ 実行制御

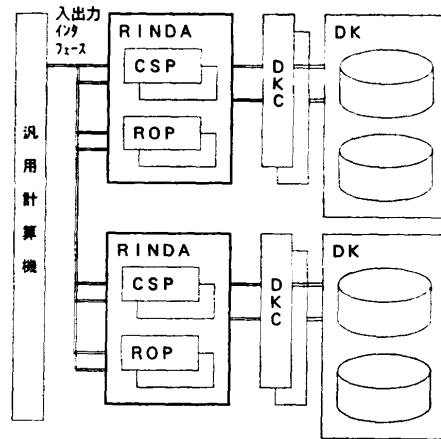
両処理の実行時の制御を共通的に行う。

#### ④ トランザクション管理機能

トランザクション内で、RINDA 処理の SQL 文とソフト処理の SQL 文が混在して使用された時の一貫性を保証するため、SQL 文の実行順序、カーソル等を共通管理する。

#### ⑤ データベース資源管理機能

同一 DB に対して、RINDA の検索処理と、ソフトウェアによる検索・更新処理が、複数ユーザから同



CSP: 内容検索プロセッサ DKC: ディスク制御装置  
ROP: 関係演算プロセッサ DK: ディスク装置

図 1 RINDA を用いたシステム構成例  
Fig. 1 Typical RINDA system organization.

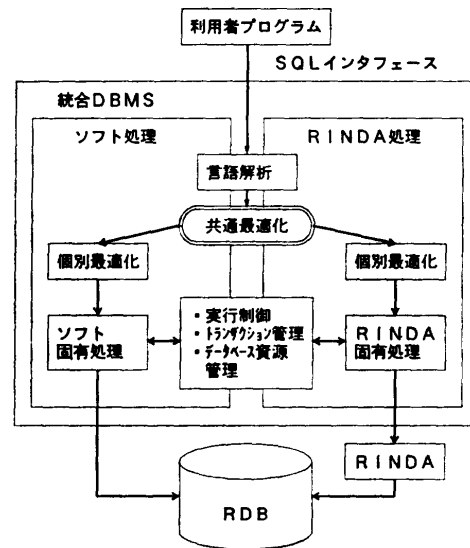


図 2 DBMS のソフトウェア構成  
Fig. 2 DBMS software organization.

時に行われた時の、トランザクション間の一貫性を保証するため、排他制御、救済制御等を共通的に行う。

## 3. 最適化の概要

RINDA 処理とソフト処理の根本的な相違により、各々独自の最適化が必要となること、ユーザが両者を共通インタフェースで利用可能とする必要があることより、最適化は共通最適化と個別最適化の 2 段階で行う方式とした。共通最適化では、機能、性能上の観点からアクセスパス選択を行い、個別最適化では、共通最適化で選択されたアクセスパスについてさらに詳細

な最適化を行う。本論文で対象とするのは、共通最適化方式である。

共通最適化は機能判定、性能判定の順で実行される。機能判定は SQL 文の種類より、定義・更新用 SQL 文はソフト処理とし、検索用 SQL 文を性能判定に引き継ぐ。性能判定では、第 5 章で示す判定基準に基づき、性能上最適なアクセスパスを決定する。性能判定基準導出の前提条件、およびその基本思想を以下に示す。

〈前提条件〉

- ① DBMS のアーキテクチャ上、SQL 文は RINDA 処理、またはソフト処理のいずれか一方で行われる。
- ② 共通最適化は、プリプロセス (SQL 文を中間言語に変換する処理) 時に行う。
- ③ 性能判定は、SQL 文の実行結果をすべてユーザへ返却するまでの処理コストで評価する。
- ④ I/O 回数と CPU 処理量はほぼ比例すると仮定する。

〈基本思想〉

- ① 原則は RINDA 処理とし、明らかに処理コストが小さいと判定できる場合のみをソフト処理とする。  
理由：非定型の検索処理に対しては、RINDA は圧倒的な速さを持つ。また、処理の大部分を専用ハードウェアで行うため汎用計算機の CPU 負荷も大幅に削減できる。
- ② 処理コストとして、I/O 回数を評価する。  
理由：より厳密な最適化のためには CPU 処理量の考慮も必要であるが、本最適化では明らかにソフト処理のコストが小さいケースを判定するため、前提条件④より I/O 回数だけの評価で十分である。
- ③ 索引の I/O 回数は処理コストから除外する。  
理由：通常、索引は主記憶上に存在することが多い。
- ④ 共通最適化は索引定義情報、ソフト処理アクセス法、および SQL 文から得られる情報のみを用いて行う。  
理由：より正確な最適化のためには、表の行数、条件に合致する行の比率等が必要であるが、これらの情報をプリプロセス時に正確に把握することは難しい。索引定義情報、ソフト処理アクセス法、SQL 文の構文情報は比較的容易に利用することができる。

4. 評価モデル

4.1 単純検索処理

(1) 格納モデル

列  $A, B$  を含む表  $P$  を仮定する。表  $P$  の行数を  $T_P$ 、表  $P$  をファイル上に連続的に格納した時のブロック数を  $B_P$  とする。列  $A, B$  の値は相互に独立かつランダムの一様分布とする。また索引の種類は、クラスタード (索引のキー順に行を連続的に格納) と、非クラスタード (索引のキー順とは無関係に行を格納) を考える。

(2) ステートメントモデル

以下の SQL 文の I/O 評価式を考える。

`select * from P where  $\phi_A$  and  $\phi_B$`

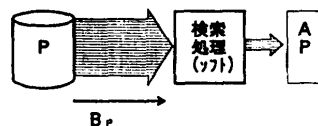
$\phi_x$ : 比較述語による列  $x$  の探索条件

$\phi_A, \phi_B$  のヒット率 (条件を満たす行数の比率) をそれぞれ  $\mu_A, \mu_B$  とし、探索条件全体のヒット率を  $\mu_P = \mu_A \mu_B$  とする。

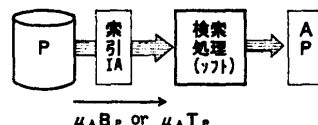
(3) I/O 評価式

(a) ソフト処理

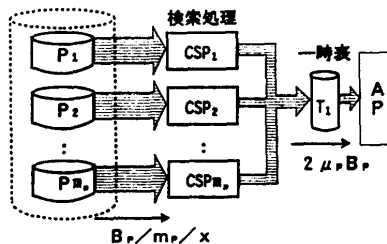
全ブロックアクセスと、索引を使用した部分ブロックアクセスの 2 種類がある。



(a) 全ブロックアクセス (ソフト処理)  
(a) All blocks access method (Software).



(b) 索引アクセス (ソフト処理)  
(b) Index access method (Software).



(c) RINDA 処理  
(c) RINDA's access method.

図 3 単純検索処理の概要

Fig. 3 Access methods for simple queries.

①全ブロックアクセス (図 3-a)

表  $P$  が格納されている全ブロック ( $B_P$  個) に I/O が必要となる。

②索引アクセス (図 3-b)

列  $A$  に付与された索引経由のアクセスを仮定する。探索条件の  $\phi_A$  を索引上で判定することによりアクセス範囲が絞り込める。I/O 数は、クラスタード索引経由では  $\mu_A B_P$ 、非クラスタード索引経由では  $\mu_A T_P$  となる。

(b) RINDA 処理 (図 3-c)

RINDA は全ブロックを専用ハードウェアで検索し、検索条件に合致した複数行をいったん一時表に蓄える。CSP による複数ディスクに分割された表の並列検索と、CSP, ROP の高速ブロックアクセス<sup>9)</sup>により、実効的に I/O 回数が削減されたのと同じ効果がある。並列検索の並列度を  $m_P$ 、高速ブロックアクセス効果を  $x$  倍とすると、表  $P$  に対する実効的な I/O 回数はソフト処理と比べて  $1/m_P/x$  倍となる。また、 $\mu_P T_P$  個の行からなる検索結果を格納する一時表サイズは  $\mu_P B_P$  となる。一時表のライト/リードを考えると、I/O 回数は  $2\mu_P B_P$  となる。

(c) 以上の結果、各アクセス方式の I/O 回数は次式で評価できる。

$$E_P(S, F) = B_P$$

$$E_P(S, CA) = \mu_A B_P$$

$$E_P(S, NA) = \mu_A T_P$$

$$E_P(R) = B_P/m_P/x + 2\mu_P B_P$$

$E_P(S, x)$ : 表  $P$  をソフト処理

$E_P(R)$ : 表  $P$  を RINDA 処理

- $x$ :  $\begin{cases} F & (\text{全ブロックアクセス}) \\ C_v & (\text{クラスタード索引アクセス}) \\ N_v & (\text{非クラスタード索引アクセス}) \end{cases}$
- $v$ : 索引が付与されている列名

4.2 ORDER BY/GROUP BY 処理

(1) 格納モデル

単純検索処理と同じとする。

(2) ステートメントモデル

```
select * from P where  $\phi_A$  and  $\phi_B$ 
order by B (または group by B)
```

(3) I/O 評価式

(a) ソフト処理

ORDER BY (OB) 句, または GROUP BY (GB) 句で指定された列に付与された索引経由で行を昇順に取り出して処理する方法 (非ソート方式) と, 探

索条件を満足する行を取り出し, 結果を一時表に蓄え, OB 句, または GB 句に指定されたキーでソート後, 昇順に取り出して処理する方法 (ソート方式) がある。

①非ソート方式 (図 4-a)

列  $B$  に付与された索引経由のアクセスを仮定する。I/O 回数は単純検索の索引アクセスと同じとなる。

②ソート方式 (図 4-b)

表  $P$  の検索部は, 単純検索のソフト処理と同じである。ソート前一時表とソート後一時表の作成と読み出し, およびソート作業用の I/O が必要となる。一時表サイズは  $\mu_P B_P$  となり, 2つの一時表の I/O 回数は  $4\mu_P B_P$  となる。また, 一般的に  $B$  個のブロック中の行ソートに必要な作業用 I/O 回数は  $B \log_2 B$  ( $k$  はマージウェイ数) となる。

(b) RINDA 処理 (図 4-c)

ソフト処理のソート方式と同じであるが, CSP, ROP により実効上の I/O 削減効果がある。表検索からソート前一時表読み出しまでの I/O 回数は, 単純検索の RINDA 処理と同じである。ROP ではソートをメモリ上で行うためソート作業用の I/O はないが, 一時表のブロック転送コストが必要となる。ROP 入出力は, 高速ブロックアクセスをしていることより, 一時

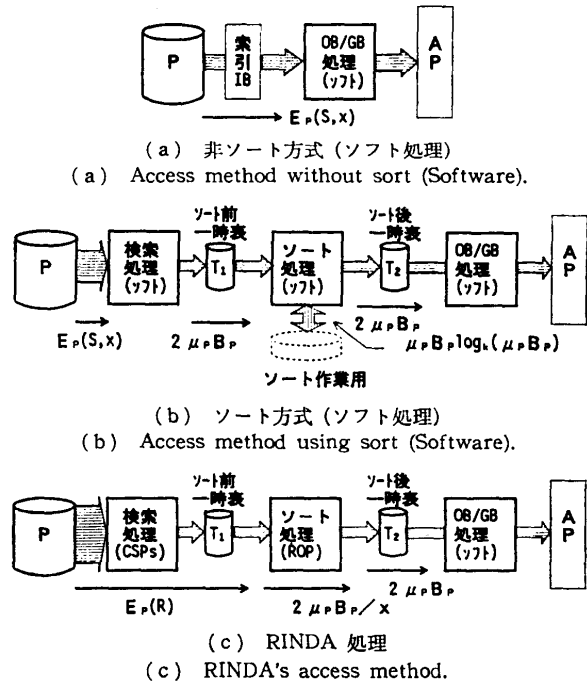


図 4 OB/GB 処理の概要  
Fig. 4 Access methods for OB or GB queries.

表の実効上の I/O 回数は、 $2\mu_P B_P/x$  となる。また、ソート後一時表に、 $2\mu_P B_P$  の I/O 回数も必要となる。

(c) 評価式

- $S_P^*(S, C_B) = E_P(S, C_B)$
- $S_P^*(S, N_B) = E_P(S, N_B)$
- $S_P(S, F) = E_P(S, F) + 4\mu_P B_P + \text{Sort}$
- $S_P(S, C_A) = E_P(S, C_A) + 4\mu_P B_P + \text{Sort}$
- $S_P(S, N_A) = E_P(S, N_A) + 4\mu_P B_P + \text{Sort}$
- $S_P(R) = E_P(R) + 2\mu_P B_P + 2\mu_P B_P/x$
- ただし  $\text{Sort} = \mu_P B_P \log_2(\mu_P B_P)$
- $S_P^*(S, x)$ : ソフト処理の非ソート方式.
- $S_P(S, x)$ : ソフト処理のソート方式.
- $S_P(R)$ : RINDA 処理
- $x$ : 単純検索処理と同じ意味

4.3 結合処理

(1) 格納モデル

列  $A, B$  を含む表  $P$  と、列  $C, D$  を含む表  $Q$  を仮定する。表  $P, Q$  の行数を  $T_P, T_Q$ 、格納ブロック数を  $B_P, B_Q$  とする。また、索引は列  $A, B, C, D$  各々にクラスタード/非クラスタードが付与された状態を考える。

(2) ステートメントモデル

$\text{select } * \text{ from } P, Q \text{ where } P.B = Q.C \text{ and } \phi_A \text{ and } \phi_B \text{ and } \phi_C \text{ and } \phi_D$

列  $x$  の探索条件  $\phi_x$  のヒット率を  $\mu_x$  とし、表  $P, Q$  のヒット率を  $\mu_P = \mu_A \mu_B, \mu_Q = \mu_C \mu_D$  とする。また、結合条件によるヒット率  $\mu_j$  を仮定する。(ただし、 $\mu_j \leq \min(\mu_B, \mu_C)$ )

(3) I/O 評価式

(a) ソフト処理

ソフト処理には、マージ結合とネステッドループ結合方式の2種類があるが、両者はほとんど同じ考え方で評価できるため、ここではマージ結合方式のみを扱うこととする。マージ結合は、両表の結合キーを各々昇順に取り出し、マージしながら結合を行う。結合キーを昇順に取り出す方法として、

- 結合キーに付与された索引を利用して取り出す。
- 結合キーでソートし取り出す。

がある。前者を索引マージ結合、後者をソートマージ結合と呼ぶこととする。

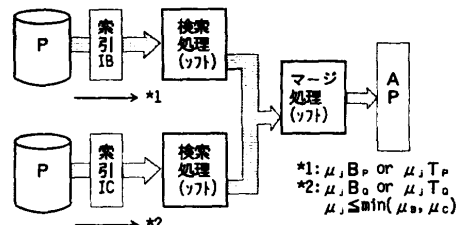
①索引マージ結合 (図 5-a)

結合キー列  $B, C$  に付与された索引を使用する。

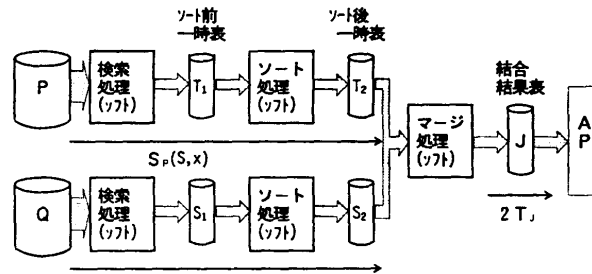
まず両索引上で結合可能な行の組を求め、次に索引経由で両表へアクセスし行の結合を行う。したがってこの方式では、結合を考慮したヒット率  $\mu_j$  が I/O 回数の支配項となる。各表の I/O 回数は、単純検索の索引アクセスと同じである。I/O 評価式は、両表の結合列に付与された索引の種類を組み合わせとなる。

②ソートマージ結合 (図 5-b)

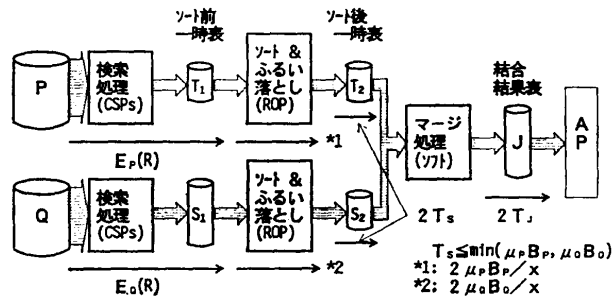
結合以外の列  $A, D$  に付与された索引を使用する。各表の検索からソート後一時表読み出までは OB/GB モデルのソート方式と同じである。結合結果表 (サイズを  $T_J$  とする) のライト/リードを考えると、 $2T_J$  の I/O 回数が必要となる。I/O 評価式は、両表のアクセス方法の組み合わせとなる。



(a) 索引マージ結合 (ソフト処理)  
(a) Index-merge join access method (Software).



(b) ソートマージ結合 (ソフト処理)  
(b) Sort-merge join access method (Software).



(c) RINDA 処理  
(c) RINDA's access method.

図 5 結合処理の概要

Fig. 5 Access methods for join query.

(b) RINDA 処理 (図 5-c)

ソートマージ結合と同じであるが, CSP, ROP により, 実効上の I/O 削減効果がある。また ROP のふるい落とし機能により, ソート後一時表のサイズが小さくなる効果もある<sup>11)</sup>。表検索からソート前一時表読み込みまでは RINDA 処理の単純検索と同じである。I/O 評価式は, ふるい落としによる I/O 削減効果も考慮する。

(c) 評価式

$$J_{IM} = \left\{ \begin{matrix} \mu_j B_P \\ \mu_j T_P \end{matrix} \right\} + \left\{ \begin{matrix} \mu_j B_Q \\ \mu_j T_Q \end{matrix} \right\}$$

$$J_{SM} = \left\{ \begin{matrix} S_P(S, F) \\ S_P(S, CA) \\ S_P(S, NA) \end{matrix} \right\} + \left\{ \begin{matrix} S_Q(S, F) \\ S_Q(S, Cd) \\ S_Q(S, Nd) \end{matrix} \right\} + 2T_J$$

$$J_R = E_P(R) + E_Q(R) + 2\mu_P B_P/x + 2\mu_Q B_Q/x + 2T_S + 2T_J$$

ただし,  $\mu_j \leq \min(\mu_B, \mu_C)$

$$T_S \leq \min(\mu_P B_P, \mu_Q B_Q)$$

$J_{IM}$ : ソフト結合処理 (索引マージ方式)

$J_{SM}$ : ソフト結合処理 (ソートマージ方式)

$J_R$ : RINDA 結合処理

$2T_S$ : ふるい落とし後のソート後一時表サイズの和

$2T_J$ : 結合結果の一時表サイズ

4.4 評価例

以上で求めた評価式を具体的に適用した例を示す。

(1) 評価条件

評価モデルで利用したものを使用する。ただし,

- $T_P = 1000$  k,  $B_P = 10$  k
- $T_Q = 1000$  k,  $B_Q = 10$  k
- 列の値は, すべてユニーク
- $m_P = m_Q = m$
- 結合処理では, 簡単のため  $\mu_P \leq \mu_Q$  を仮定

(2) 評価例

ヒット率  $\mu_P$  を横軸, ソフト処理, および RINDA 処理の I/O 回数を縦軸にとり, 両処理を比較した。

単純検索処理の計算結果を図 6, OB/GB 処理の計算結果を図 7 に示す。また結合処理の計算結果は, 索引マージと, ソートマージに分け,

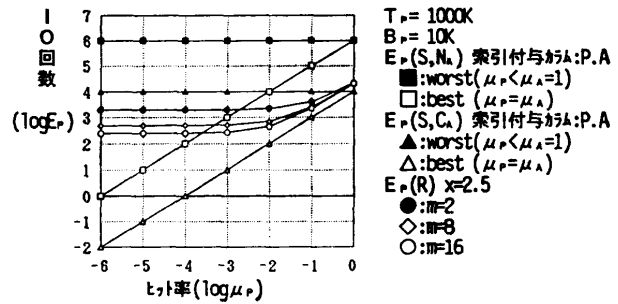


図 6 単純検索処理の IO 回数  
Fig. 6 Number of IOs in simple queries.

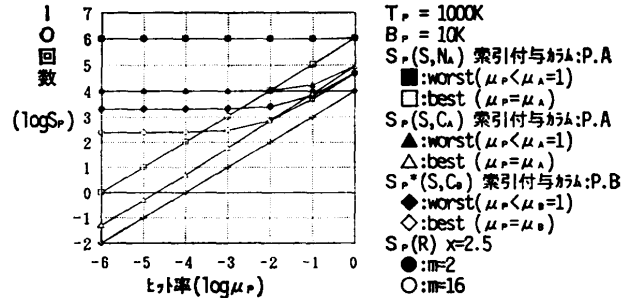
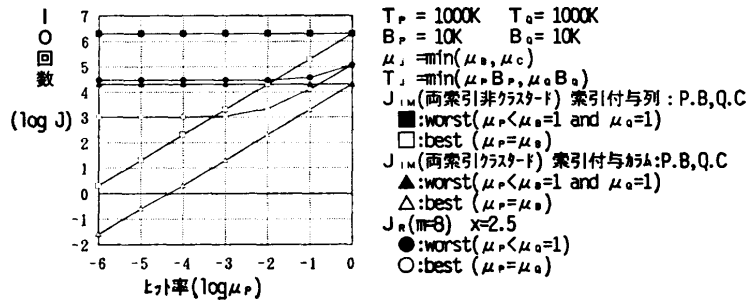
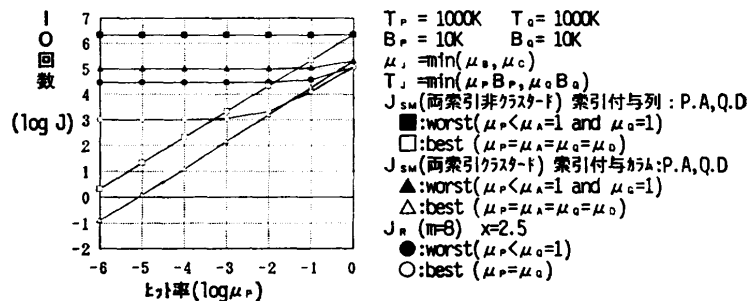


図 7 OB/GB 処理の IO 回数  
Fig. 7 Number of IOs in OB or GB queries.



(a) 索引マージ結合 (ソフト処理) と RINDA 処理  
(a) Index-merge join access method (Software) and RINDA's access method.



(b) ソートマージ結合 (ソフト処理) と RINDA 処理  
(b) Sort-merge join access method (Software) and RINDA's access method.

図 8 結合処理の IO 回数  
Fig. 8 Number of IOs in join queries.

前者を図 8-a, 後者を図 8-b に示す.

### 5. アクセスバス判定基準の決定

4章で求めた I/O 評価考を元に, ソフト処理の I/O 回数が RINDA 処理と比べて明らかに小さいと判定できる十分条件を求め, ソフト処理を選択するための判定基準を決定する.

#### 5.1 単純検索処理

- (1)  $E_P(S, F) < E_P(R)$  となる十分条件

上式を整理すると,  $1 < 1/m_P/x + 2\mu_P$

$\mu_P=1$  の時, 明らかに,  $E_P(S, F) < E_P(R)$

- (2)  $E_P(S, C_A) < E_P(R)$  となる十分条件

上式を整理すると,  $\mu_A < 1/m_P/x + 2\mu_P$

(a)  $\mu_P=1$  の時, 明らかに  $E_P(S, C_A) < E_P(R)$

(b)  $\mu_P=\mu_A$  の時, 明らかに  $E_P(S, C_A) < E_P(R)$

(c)  $1/m_P/x = 10^{-2}$  程度を仮定すると,

$\mu_A < 10^{-2}$  の時,  $\mu_A < 10^{-2} \doteq 1/m_P/x$

したがって,  $E_P(S, C_A) < E_P(R)$  が成立する.

- (3)  $E_P(S, N_A) < E_P(R)$  となる十分条件

上式を整理すると

$\mu_A T_P/B_P < 1/m_P/x + 2\mu_P$

$1/m_P/x = 10^{-2}$ ,  $T_P/B_P = 10^2$  程度を仮定すると,

$\mu_A < 10^{-4}$  の時,

$\mu_A T_P/B_P < 10^{-2} \doteq 1/m_P/x$

したがって,  $E_P(S, N_A) < E_P(R)$  が成立する.

- (4) 判定基準の決定

上記の十分条件を元に, SQL 文から抽出可能な条件を求めそれを用いて判定基準を決定した. 十分条件は① $\mu_P=1$ , ② $\mu_P=\mu_A$ , ③ $\mu_P \leq \mu_A < \alpha$  の3種類に分類できる. 各々に対応して次の3つの SQL 文抽出可能条件を設定した.

**条件 1: where 句に探索条件がない.**

$\mu_P=1$  は全行ヒットを意味する. SQL 文のみから得られる条件として上記を設定した.

**条件 2: where 句に探索条件のみが指定されている.**

$\mu_P=\mu_A$  は探索条件全体のヒット率が索引上の探索条件のヒット率に等しいことを意味する. これは, 索引に適合する探索条件のみが指定されていると考え, SQL 文のみから得られる条件として上記を設定した.

**条件 3: where 句に「列名=定数またはホスト変数」の探索条件が指定されている.**

$\mu_P \leq \mu_A < \alpha$  は索引でのヒット率がある値以下であることを意味する. しかし十分条件の導出例でもわかるように  $\alpha$  は索引の種類で異なる. また, 最適化時

に正確なヒット率を得るのも困難である. そこで, 「キー値の重複度が小さい列に索引が付与される」という経験的事実と, 「重複度が小さい列に対する等号条件のヒット率は十分小さい」という仮定より, SQL 文のみから得られる情報として上記を設定した.

#### 【単純検索処理の判定基準】

ソフト処理のアクセス法と SQL 文抽出可能条件を用い, ソフト処理とする以下の判定基準を決定した.

- (A) ソフト処理が全ブロックアクセスの場合  
探索条件が【条件 1】を満足
- (B) ソフト処理がクラスタード索引アクセスの場合  
索引に適用可能な探索条件が【条件 1】 or 【条件 2】 or 【条件 3】を満足
- (C) ソフト処理が非クラスタード索引アクセスの場合  
索引に適用可能な探索条件が【条件 3】を満足

#### 5.2 OB/GB 処理

ソフト処理のアクセスのうち, 全ブロックアクセスは, 明らかにソフト処理の I/O 回数が大きくなるため, 索引経由のアクセスについて, その十分条件を求める.

- (1)  $S_P^*(S, C_B) < S_P(R)$  となる十分条件

上式を整理すると

$\mu_A < 1/m_P/x + 4\mu_P + 2\mu_P/x$

(a)  $\mu_P=1$  の時, 明らかに  $S_P^*(S, C_B) < S_P(R)$

(b)  $\mu_A=\mu_P$  の時, 明らかに  $S_P^*(S, C_B) < S_P(R)$

(c)  $1/m_P/x = 10^{-2}$  程度を仮定すると,

$\mu_A < 10^{-2}$  の時,

$\mu_A < 10^{-2} \doteq 1/m_P/x$

より,  $S_P^*(S, C_B) < S_P(R)$  が成立する.

- (2)  $S_P^*(S, N_B) < S_P(R)$  となる十分条件

上式を整理すると

$\mu_A T_P/B_P < 1/m_P/x + 4\mu_P + 2\mu_P/x$

$1/m_P/x = 10^{-2}$ ,  $T_P/B_P = 10^2$  程度を仮定すると,

$\mu_A < 10^{-4}$  の時,

$\mu_A T_P/B_P < 10^{-2} \doteq 1/m_P/x$

より,  $S_P^*(S, N_B) < S_P(R)$  が成立する.

- (3)  $S_P(S, C_A) < S_P(R)$  となる十分条件

上式を整理すると

$\mu_A + \mu_P \log_2(\mu_P B_P) < 1/m_P/x + 2\mu_P/x$

$\mu_P < \mu_A$  より

$\mu_A + \mu_P \log_2(\mu_P B_P) < \mu_A + \mu_A \log_2(\mu_A B_P)$

$1/m_P/x=10^{-2}$ ,  $k=10$  程度,  $B_P < 10^{10}$  を仮定すると,  $\mu_A < 10^{-3}$  の時

$$\mu_A + \mu_A \log_k(\mu_A B_P) < 10^{-2} \cong 1/m_P/x$$

より,  $S_P(S, C_A) < S_P(R)$  が成立する.

(4)  $S_P(S, N_A) < S_P(R)$  となる十分条件

上式を整理すると

$$\begin{aligned} \mu_A T_P/B_P + \mu_P \log_k(\mu_P B_P) \\ < 1/m_P/x + 2\mu_P/x \end{aligned}$$

$\mu_P < \mu_A$  より,

$$\begin{aligned} \mu_A T_P/B_P + \mu_P \log_k(\mu_P B_P) \\ < \mu_A T_P/B_P + \mu_A \log_k(\mu_A B_P) \end{aligned}$$

$1/m_P/x=10^{-2}$ ,  $k=10$  程度,  $B_P < 10^{10}$  を仮定すると,  $\mu_A < 10^{-5}$  の時

$$\begin{aligned} \mu_A T_P/B_P + \mu_A \log_k(\mu_A B_P) \\ < 10^{-2} \cong 1/m_P/x \end{aligned}$$

よって,  $S_P(S, N_A) < S_P(R)$  が成立する.

(5) 判定基準の決定

上記の十分条件を整理すると単純検索と同じ分類となる。したがって単純検索で設定した **SQL 文抽出可能条件**を利用して判定基準を決定した。

#### 【OB/GB 処理の判定基準】

ソフト処理とする判定基準

(A) ソフト処理が非ソート方式の場合

(a) クラスタード索引アクセスの場合  
索引に適用可能な探索条件が **【条件1】** or **【条件2】** or **【条件3】** を満足

(b) 非クラスタード索引アクセスの場合  
索引に適用可能な探索条件が **【条件3】** を満足

(B) ソフト処理がソート方式の場合

索引アクセスで, 索引に適用可能な制限条件が **【条件3】** を満足

#### 5.3 結合処理

索引マージ結合, ソートマージ結合各々について, 判定基準を決定する.

##### 5.3.1 索引マージ結合の場合

簡単化のため, 結合列に付与された索引の種類は同じとした。また,  $\mu_B \leq \mu_C$  となるように表  $P, Q$  を選ぶ。この条件で,  $J_{IM} < J_R$  となる十分条件を求める。

(1) 両索引が共にクラスタードの場合

$$J_{IM} = \mu_j B_P + \mu_j B_Q$$

を考える。  $\mu_j \leq \min(\mu_B, \mu_C)$  の関係,  $\mu_B \leq \mu_C$  を仮定すると

$$J_{IM} \leq \mu_B B_P + \mu_B B_Q$$

となる。ここで  $J_{IM} < J_R$  を次のように整理する。

$$\mu_B B_P + \mu_B B_Q < 2\mu_P B_P + 2\mu_Q B_Q + \dots$$

(a)  $\mu_P = \mu_Q = 1$  の時, 明らかに  $J_{IM} < J_R$

(b)  $\mu_B = \mu_P$  and  $\mu_C = \mu_Q$  の時

$$\begin{aligned} \mu_B B_P + \mu_B B_Q \\ < \mu_B B_P + \mu_C B_Q \quad ( \because \mu_B < \mu_C ) \end{aligned}$$

$$= \mu_P B_P + \mu_Q B_Q$$

$$< 2\mu_P B_P + 2\mu_Q B_Q + \dots$$

となり,  $J_{IM} < J_R$  が成立する。

(c)  $J_{IM} < J_R$  を以下のように再整理すると

$$\mu_B B_P + \mu_B B_Q$$

$$< B_P/m_P/x + B_Q/m_Q/x + \dots$$

$1/m_P/x, 1/m_Q/x$  が共に  $10^{-2}$  程度を仮定する。  $\mu_B < 10^{-2}$  の時

$$\mu_B < 10^{-2} \cong 1/m_P/x$$

$$\mu_B < 10^{-2} \cong 1/m_Q/x$$

よって本条件は  $J_{IM} < J_R$  の十分条件となる。

(2) 両索引が共に非クラスタードの場合

$$J_{IM} = \mu_j T_P + \mu_j T_Q$$

を考える。  $\mu_j \leq \min(\mu_B, \mu_C)$  の関係, および  $\mu_B < \mu_C$  を仮定すると

$$J_{IM} \leq \mu_B T_P + \mu_B T_Q$$

となる。上記条件を用いて  $J_{IM} < J_R$  を整理すると

$$\mu_B T_P + \mu_B T_Q$$

$$< B_P/m_P/x + B_Q/m_Q/x + \dots$$

$1/m_P/x, 1/m_Q/x$  が共に  $10^{-2}$  程度, また  $T_P/B_P, T_Q/B_Q$  が共に  $10^2$  程度を仮定すると,  $\mu_B < 10^{-4}$  の時, 以下が成立する。

$$\mu_B T_P/B_P < 10^{-2} \cong 1/m_P/x$$

$$\mu_B T_Q/B_Q < 10^{-2} \cong 1/m_Q/x$$

よって, 本条件は,  $J_{IM} < J_R$  の十分条件となる。

なお,  $J_{IM}$  (片方が非クラスタード索引)  $< J_{IM}$  (両方が非クラスタード索引) は自明より, 本条件は少なくとも片方が非クラスタード索引でも十分条件となる。

(3) 判定基準の決定

上記の十分条件を整理すると単純検索と同じ分類となる。したがって単純検索で設定した **SQL 文抽出可能条件**を利用して決定した。

#### 【索引マージ結合処理の判定基準】

ソフト処理とする判定基準

(A) 両表をクラスタード索引アクセスする場合

索引に適用可能な探索条件が **【条件1】** or **【条件2】** or **【条件3】** を満足



(B) 一方の表を非クラスタード索引アクセスする場合

索引に適用可能な探索条件が[条件3]を満足

5.3.2 ソートマージ結合の場合

I/O 回数が大きくなる全ブロックアクセスを含む組み合わせは除外し、索引アクセスの組み合わせのみを扱う。また使用する両索引は同じものと仮定する。これらの条件より、 $J_{SM} < J_R$  となる十分条件を求める。

(1) 両索引がクラスタードの場合

$$J_{SM} = S_P(S, C_A) + S_Q(S, C_D) + 2T_I$$

を考える。 $J_{SM} < J_R$  を整理すると

$$\begin{aligned} &\mu_A B_P + 2\mu_P B_P + \mu_P B_P \log_k(\mu_P B_P) \\ &+ \mu_D B_Q + 2\mu_Q B_Q + \mu_Q B_Q \log_k(\mu_Q B_Q) \\ &< B_P/m_P/x + B_Q/m_Q/x + 2T_S \end{aligned}$$

上式が成立する十分条件は、

$$\mu_A + 2\mu_P + \mu_P \log_k(\mu_P B_P) < 1/m_P/x \quad (\alpha 1)$$

$$\mu_D + 2\mu_Q + \mu_Q \log_k(\mu_Q B_Q) < 1/m_Q/x \quad (\beta 1)$$

$1/m_P/x = 10^{-2}$ ,  $k=10$  程度,  $B_P < 10^{10}$  を仮定すると、 $\mu_A < 10^{-3}$  の時、

$$\begin{aligned} &\mu_A + 2\mu_P + \mu_P \log_k(\mu_P B_P) \\ &< 3\mu_A + \mu_A \log_k(\mu_A B_P) \quad (\because \mu_A > \mu_P) \\ &< 10^{-2} \cong 1/m_P/x \end{aligned}$$

よって、この条件の時、式 (α1) が成り立つ。同様に、 $\mu_D < 10^{-3}$  の時、式 (β1) が成り立つ。以上より、 $\mu_A < 10^{-3}$ ,  $\mu_D < 10^{-3}$  の時、 $J_{SM} < J_R$  が成立する。

(2) 両索引が非クラスタードの場合

$$J_{SM} = S_P(S, N_A) + S_Q(S, C_D) + 2T_I$$

を考える。 $J_{SM} < J_R$  を整理すると

$$\begin{aligned} &\mu_A T_P + 2\mu_P B_P + \mu_P B_P \log_k(\mu_P B_P) \\ &+ \mu_D T_Q + 2\mu_Q B_Q + \mu_Q B_Q \log_k(\mu_Q B_Q) \\ &< B_P/m_P/x + B_Q/m_Q/x + 2T_S \end{aligned}$$

上式が成立する十分条件は、

$$\mu_A T_P/B_P + 2\mu_P + \mu_P \log_k(\mu_P B_P) < 1/m_P/x \quad (\alpha 2)$$

$$\mu_D T_Q/B_Q + 2\mu_Q + \mu_Q \log_k(\mu_Q B_Q) < 1/m_Q/x \quad (\beta 2)$$

$1/m_P/x = 10^{-2}$ ,  $T_P/B_P = 10^{-2}$ ,  $k=10$  程度,  $B_P < 10^{10}$  を仮定すると、 $\mu_A < 10^{-5}$  の時、

$$\begin{aligned} &\mu_A T_P/B_P + 2\mu_P + \mu_P \log_k(\mu_P B_P) \\ &< \mu_A T_P/B_P + 2\mu_A + \mu_A \log_k(\mu_A B_P) \quad (\because \mu_A > \mu_P) \\ &< 10^{-2} \cong 1/m_P/x \end{aligned}$$

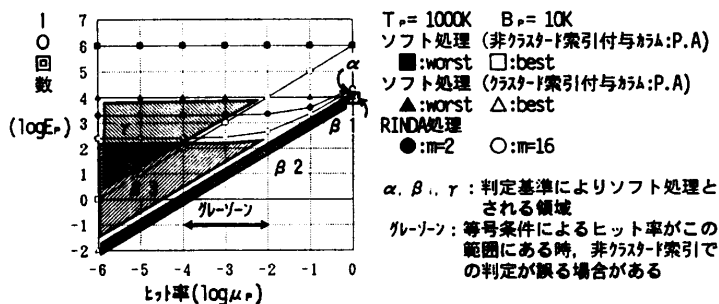


図 9 単純検索処理への判定基準適用例  
Fig. 9 Access path selection to simple queries.

よって、この条件の時、式 (α2) が成り立つ。同様に、 $\mu_D < 10^{-5}$  の時、式 (β2) が成り立つ。以上より、 $\mu_A < 10^{-5}$ ,  $\mu_D < 10^{-5}$  の時、 $J_{SM} < J_R$  が成立する。

(3) 判定基準の決定

上記十分条件はすべて単純検索の  $\mu_P \leq \mu_A < \alpha$  の場合であり、単純検索で設定した SQL 文抽出可能条件を利用して判定基準を設定した。

【ソートマージ結合処理の判定基準】

ソフト処理とする判定基準

(A) 両表を共に索引アクセスする場合

両索引各々に対し適用可能な探索条件が[条件3]を満足

5.4 判定基準適用例

4.4 節で示した単純検索の I/O 回数結果例 (図 6) に対して判定基準を適用した例を図 9 に示す。判定基準 A, B, C によりソフト処理と判定される領域を  $\alpha, \beta$  ( $\beta 1 \sim \beta 3$ ),  $\gamma$  で示す。 $\beta 1 \sim \beta 3$  は判定基準 B の条件 1~3 に対応している。特に  $\beta 3$  および  $\gamma$  は索引に適用可能な等号条件のヒット率に判定基準の正誤が左右されることを示している。例えば等号条件のヒット率が  $10^{-3}$  の時、 $\beta 3$  での判定は正しいが、 $\gamma$  での判定は誤る。しかし、5.1 節の SQL 文抽出可能条件 3 で述べたように「索引が付与された列に対する等号条件のヒット率は十分小さい」ことを考えると判定を誤るケースは少ないと考える。また、すべての判定基準は、索引定義、SQL 文情報のみから求めているため、判定を誤る場合もある。例えば、範囲条件でも指定範囲が十分小さく等号条件と同等のヒット率となると判定を誤る。しかし、この場合、汎用計算機の CPU 負荷を大幅に削減する RINDA 処理となるため大きな問題にはならないと考える。

## 6. ま と め

データベースプロセッサ RINDA とソフトウェアの DB 処理を統合した DBMS のアクセスパス決定方式について示した。アクセスパス決定は、原則は RINDA 処理とし、十分少ない I/O 回数で処理可能な場合のみ、ソフト処理とする方式である。その判定基準は、処理をモデル化して求めた I/O 評価式から、ソフト処理の I/O 回数が小さくなる十分条件を導き、ソフト処理のアクセス方法、索引定義情報、SQL 文情報を用いて決定した。判定基準概要を以下に示す。

### (1) 単純検索処理

- 索引に適用可能でかつ I/O 削減効果の大きい探索条件が指定されている。

### (2) OB/GB 処理

- ソフト処理方式を非ソート方式、ソート方式に分類し、各々において、索引に適用可能でかつ I/O 削減効果の大きい探索条件が指定されている。

### (3) 結合処理

- ソフト処理方式を索引マージ方式、ソートマージ方式に分類し、索引に適用可能でかつ I/O 削減効果の大きい探索条件が指定されている。
- 本文中では述べなかったが、ネステッドループ方式を用いる場合にも上記と同様な考え方で判定基準を設定できる。

なお、上記の判定基準は索引定義情報、SQL 文情報のみから求めているため、ソフト処理の方が速い場合をすべて網羅できていない。しかし簡単に取得できる情報のみを用いてはば満足できる最適化を実現している点で実用的な方法であると言える。

**謝辞** 本研究の機会を与えてくださった NTT 情報通信網研究所松永俊雄主席研究員、拜原正人情報処理研究部長に感謝するとともに、有益なご指導、ご助言を与えてくださった佐藤哲司主任研究員に深く感謝いたします。

## 参 考 文 献

- 1) 速水, 井上ほか: 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, 情報処理学会計算機アーキテクチャ研究会資料, 88-ARC-73-12 (1988).
- 2) Inoue, U., Hayami, H., Fukuoka, H. and Suzuki, K.: RINDA—A Relational Database Processor for Non-Indexed Queries, *Proc. Int. Symp. Database Systems for Advanced Applications*, pp. 382-386 (1989).

- 3) 井上, 速水ほか: データベースプロセッサ RINDA の設計の実現, 情報処理学会論文誌, Vol. 31, No. 3, pp. 373-380 (1990).
- 4) 黒岩, 板倉ほか: データベースプロセッサ RINDA の性能向上効果, 第 39 回情報処理学会全国大会論文集, pp. 1104-1105 (1989).
- 5) 井上, 芳西ほか: データベースプロセッサ RINDA の制御方式, 信学技法, DE 89-41, pp. 41-47 (1989).
- 6) JIS X 3005 データベース言語 SQL (1989).
- 7) Blasgen, M. W. and Eswaran, K. P.: Storage and Access in Relational Data Bases, *IBM Syst. J.*, Vol. 16, No. 4, pp. 363-377 (1977).
- 8) Selinger, P. G., Astrahan, M. M., Chamberlin, D. D., Lorie, R. A. and Price, T. G.: Access Path Selection in a Relational Database Management System, *Proc. of ACM-SIGMOD*, pp. 23-34 (1979).
- 9) 速水, 武田ほか: データベースプロセッサ RINDA の内容検索方式, 第 37 回情報処理学会全国大会論文集, pp. 379-380 (1988).
- 10) 武田, 佐藤ほか: 関係演算高速化プロセッサ, 情報処理学会論文誌, Vol. 31, No. 8, pp. 1230-1241 (1990).
- 11) 中村, 板倉ほか: データベースプロセッサ RINDA の最適化方式, 第 37 回情報処理学会全国大会論文集, pp. 385-386 (1988).

(平成 2 年 9 月 26 日受付)

(平成 3 年 9 月 12 日採録)



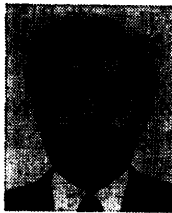
芳西 崇 (正会員)

昭和 32 年生。昭和 56 年九州工業大学電子工学科卒業。昭和 58 年東京工業大学大学院総合理工学研究科修士課程修了。同年、日本電信電話公社入社。現在、NTT 情報通信網研究所データベース研究部主任研究員。データベース管理システム、データベースマシンの研究実用化に従事。電子情報通信学会会員。

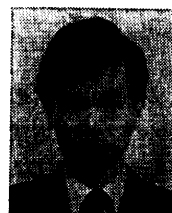


板倉 一郎 (正会員)

昭和 34 年生。昭和 56 年電気通信大学電気通信学部計算機科学科卒業。昭和 58 年同大学院電気通信学研究科計算機科学専攻修士課程修了。同年、日本電信電話公社入社。現在、NTT 情報通信網研究所情報処理研究部主任研究員。主に、データベースマシンの研究実用化に従事。電子情報通信学会会員。

**中村 敏夫 (正会員)**

昭和24年生。昭和47年京都工芸繊維大学工学部電気工学科卒業。同年、日本電信電話公社入社。現在、NTT 情報通信網研究所データベース研究部主任研究員。主に、言語処理の研究実用化を行い、現在データベースマシンの研究実用化に従事。電子情報通信学会会員。

**井上 潮 (正会員)**

昭和28年生。昭和50年名古屋大学工学部電気学科卒業。同年、日本電信電話公社入社。現在、NTT 情報通信網研究所データベース研究部主幹研究員。オンライン情報検索システム、データベース管理システム、データベースマシンの研究実用化に従事。電子情報通信学会、IEEE-CS 各会員。