

# データベースプロセッサ RINDA の 4N-2 不可視化制御方式

芳西 崇      板倉 一郎      中村 敏夫

NTT 情報通信処理研究所

## 1. はじめに

データベースプロセッサ RINDA<sup>[1]</sup> はリレーショナルデータベース処理の性能向上を目的として開発された専用プロセッサである。利用者は、SQL<sup>[2]</sup> 文により RINDA アクセスとソフトウェアによるアクセス(非 RINDA アクセス)を同一 AP で実行可能であり<sup>[3]</sup>、RINDA を使用するか否かを SQL 文毎に指定することができる。本稿では利用者側でのこの指定を不要とし、性能的に優れるアクセスパスをデータベース管理システム(DBMS)が自動的に決定する不可視化制御方式について述べる。

## 2. 現状の問題点

SQL 文毎に RINDA 使用有無を指定することにより利用者側はアクセスの最適化を選択することができる。その場合、(1) RINDA の高速な検索機能を利用するために SQL 文が検索系であるか、(2) 検索系 SQL 文は性能的に RINDA / 非 RINDA アクセスのどちらが優れるか、を選択する必要がある。特に性能上の選択は、効果的にインデックスを利用する場合と RINDA を利用する場合との性能の比較を SQL 文毎に判定する必要があり、利用者には困難な場合がある。

## 3. RINDA 不可視化制御方式

SQL 文毎の RINDA 指定を不要とすることにより利用者の RINDA 使用の意識を無くし(不可視化)、DBMS が SQL 文から RINDA / 非 RINDA アクセスを判定し(最適化)、実行する方式を実現した。本稿では特に最適化方式について述べる。

### 3.1 最適化方式

SQL 文から機能及び性能上の観点より RINDA / 非 RINDA アクセスパスを決定する。

### 3.2 前提条件

- ① RINDA / 非 RINDA のアクセスパスはプリプロセス時に決定する。
- ② インデックスの付与状態と条件式の種類のみを最適化条件とする。(ヒット率、行総数等の統計情報は考慮しない。)
- ③ 性能上のアクセスパス判定は、条件を満足する行を全て返却するまでの I/O 回数とする。

### 3.3 アクセスパス判定方式

機能、性能の観点の順でアクセスパスを判定する。詳細を以下に述べる。

#### (1) 機能上のアクセスパス判定方法

RINDA は検索系処理のみを実行するため検索系以外の SQL 文は非 RINDA アクセスとする。

#### (2) 性能上のアクセスパス判定方法

両アクセスで処理可能な検索系 SQL 文について以下の性能上の観点からそのアクセスパスを判定する。

① RINDA が性能的に必ず優れる機能が指定されている場合、RINDA アクセスとする。

RINDA は表 1 に示すデータベース処理機能を専用ハード化しており<sup>[4][5]</sup>、SQL 文にこれらの機能が SQL 文に指定されていれば RINDA が必ず性能的に優れる。

RINDA ハード化機能	SQL 文指定
文字列検索機能	LIKE 述語
行数計数機能	COUNT(*)
重複除去機能	DISTINCT

表 1 RINDA ハード化機能

従って以下の条件を満たす時 RINDA アクセスとする。

<条件 I>

SQL 文に、① LIKE 述語、② COUNT(\*) 関数  
③ DISTINCT 指定、が少なくとも 1 つ指定されている

② 非 RINDA アクセスが十分少ない I/O 回数で処理可能な場合、非 RINDA アクセスとする。

RINDA / 非 RINDA アクセス処理方式の相違を図 1 に示す。RINDA は表をフルサーチで高速に検索する。一方非 RINDA アクセスではインデックスが有効に利用できるアクセス対象行が絞り込める場合には少ない I/O 回数で処理が可能となる。

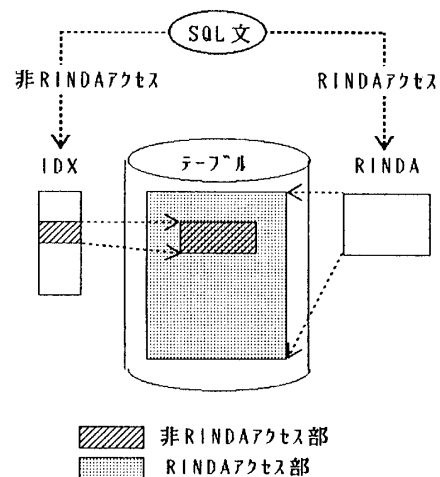


図 1 RINDA と非 RINDA アクセス処理方式

#### (i) 結合が無い場合

インデックス・キー値の条件で対象行が十分絞り込めれば I/O 回数を減らすことができ、非 RINDA アクセスが良い。従って以下の条件を満たす場合非 RINDA アクセスとする。

＜条件Ⅱ＞  
 ① WHERE 句に OR 条件が指定されていない。  
 ② WHERE 句に以下の指定がある。  
 列名 = ホスト変数 / 定数  
 ③ ②を満足する列にインデックスが付与され、非 RINDA がこのインデックスを使用してアクセスする。

結合がある場合  
 非 RINDA の結合処理は①マージ結合、②ネステッドループ結合方式を採用している。

基本的には結合処理は RINDA 利用により RINDA アクセスが高速 (図 2-1) である。しかし、非 RINDA の結合処理で次のようなインデックス利用をすれば少ない I/O 回数で処理が可能となり、非 RINDA アクセスで良い。

(a) マージ結合 (図 2-2)

- 条件を満足する行検索が、" = " 条件で且つインデックス・キー値で絞りこめれば、I/O 回数を少なくできる。
- 絞り込んだ行数が十分少なければソート処理は十分小さい。

(b) ネステッドループ結合 (図 2-3)

- 主側の表の条件検索が " = " 条件で且つインデックス・キー値で絞り込めれば I/O 回数を少なくできる。
- 従側の表の結合列にインデックスが付与されていれば、主側の行に対応する従側の結合行が少ない I/O 回数でヒットできる。

以上より結合処理を非 RINDA アクセスとする新たな条件を以下に示す。

＜マージ結合＞

＜条件Ⅲ＞  
 ① 結合する表各々について条件Ⅱを満たす条件が指定されている。

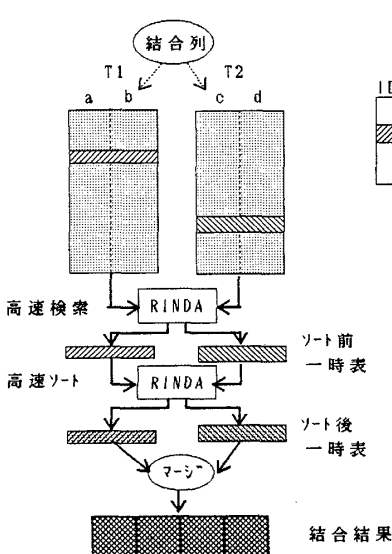


図 2-1 RINDA 結合処理

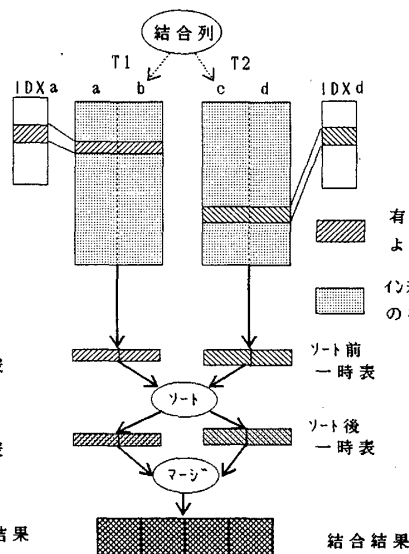


図 2-2 非 RINDA マージ結合処理

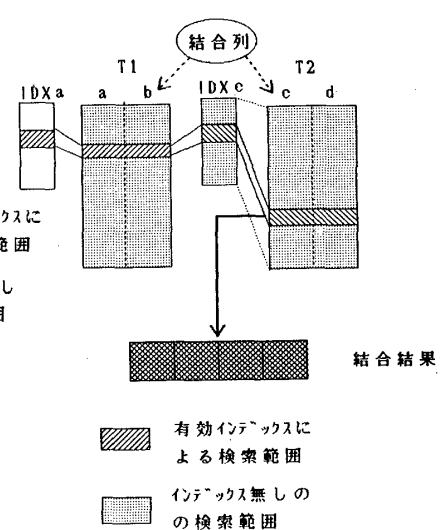


図 2-3 非 RINDA ネステッドループ結合処理

＜ネステッドループ結合＞

＜条件Ⅳ＞  
 ① 主側の表の結合列以外の列に条件Ⅱを満たすものがある。  
 ② 従側の表の結合列にインデックスが付与されている。

(3) 性能上の RINDA / 非 RINDA アクセス判定  
 上記の性能上の観点によるアクセスパス判定条件を表 2 にまとめる。

結合		条件				T: 真 F: 偽 -: ケース無
		I	II	III	IV	
有	マージ	F	T	-	-	
	ネステッド	F	T	-	T	
無		F	T	-	-	

表 2 非 RINDA アクセスとするケース

注) 全ての条件が上記の場合非 RINDA アクセス

4. おわりに

利用者から RINDA を不可視とし、DBMS が RINDA / 非 RINDA アクセスパスを決定する最適化方式を実現した。これにより、利用者は RINDA を意識せず性能的に優れた SQL 文の実行が可能となった。

今回の不可視化制御の最適化では、インデックス付与状態、条件式の種類のみの情報を利用しているため、ヒット率、表の総行数等で最適化が誤るケースもある。今後現状の最適化の正当性の検証を行なうとともに、統計情報等を利用した最適化的的中率を向上させる方式を検討して行く予定である。

【参考文献】

[1] 井上他「データベース・リセッサ RINDA のアーキテクチャ」, 情報処理学会第 37 回全国大会 5Q-4, 1988  
 [2] JIS X 3005 データベース言語 SQL, 1987  
 [3] 芳西他「データベース・リセッサ RINDA の問合せ処理方式」, 情報処理学会第 37 回全国大会 5Q-7, 1988  
 [4] 速水他「データベース・リセッサ RINDA の内容検索方式」, 情報処理学会第 37 回全国大会 5Q-5, 1988  
 [5] 武田他「データベース・リセッサ RINDA の関係演算方式」, 情報処理学会第 37 回全国大会 5Q-6, 1988