

3Q-2

# データベースプロセッサRINDAの副問合せ最適化方式

板倉 一郎、 芳西 崇、 中村 仁之輔  
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[1]は、リレーショナルデータベース処理の性能向上を目的に開発された専用プロセッサで、CSP(内容検索プロセッサ)とROP(関係演算プロセッサ)から構成される。RINDAを制御するデータベース管理システム(DBMS)では、検索実行時に事前の処理結果を参照してアクセスパスを決定する動的最適化方式を採用している。本稿では、リレーショナルデータベース言語SQL[2]の限定述語に関してDBMSが行っている最適化方式について述べる。

## 2. 動的最適化方式

DBMSは次のように検索を実行する。  
 ①CSPで表をサーチし、結果を一時表(検索の途中結果を格納する作業用の表)に格納する。  
 ②一時表に対して副問合せ・結合・ソート等の演算をCPUまたはROPで行い、結果を一時表に格納する。  
 この検索方式の特徴は一時表の情報(行数、ページ数等)をDBMSが実行時に参照できることである。この点に着目し、副問合せ・結合のように複数の表を演算する場合、一時表の情報を利用して検索対象の表へのアクセスパス・演算方法をDBMSは実行時に決定している。これを動的最適化方式[3]と言い、効率の良いアクセスパス・演算方法を選択でき、検索処理時間を短縮することができる。

## 3. 副問合せ最適化方式

### 3.1 従来方式の問題点

[検索命令]

```
SELECT *
FROM 会社
WHERE 地区 = '東京' AND 社番 = ANY (SELECT 社番
FROM 部品庫
WHERE 在庫 (<= 50))
```

[データの流れ]

検索結果

社番	社名	地区	電話
3	B社	東京	03-xxx-
2	D社	東京	03-xxx-
⋮			
⋮			

会社表

社番	社名	地区	電話番号
1	A社	横浜	045-XXX-
3	B社	東京	03-xxx-
4	C社	東京	03-xxx-
2	D社	東京	03-xxx-
⋮			
⋮			

[処理の流れ]

- ①部品表の検索  
条件: 在庫 (<= 50)
- ②会社表の検索  
条件: 地区 = '東京'  
を満足する各行について社番が作業域に存在するか比較

作業域

社番
5
3
2
10
⋮
⋮

部品表

品番	品名	在庫	価格	社番
E 1	電池	40	90	5
S 1	ソケット	50	180	3
E 2	電球	20	150	2
S 2	スイッチ	100	480	2
S 3	ネジ	30	180	10
⋮				
⋮				

図1 従来方式による副問合せ処理方式例

従来のDBMSはソフトウェアにより限定述語を以下のように処理した。(図1)

- ①副問合せ側の表を検索し、結果を作業域に格納する。
- ②主問合せ側の表で条件を満足する各行について、作業域から副問合せ結果を1件ずつ取り出し限定述語の条件を満足するか判定する。

この方法では、副問合せ結果の要素値全てに対して、比較演算子の条件を満足するかどうか判定するため、主問合せの表の行数をM、副問合せの結果の要素値の件数をNとすると、限定述語の判定時間はM\*Nのオーダーとなり、処理時間が長くなるという問題があった。

### 3.2 RINDA利用の着眼点

限定述語の判定には、列の値(主問合せの表)と複数の値(副問合せ結果)との比較が必要である。このため、RINDAの以下の機能を利用することにより高速に処理することが可能である[4][5]。

- ①CSPのIN述語判定機能を用いて、列の値と複数の値を比較する。
- ②主問合せの列と副問合せ結果の比較回数を減らすためには、両者をソートしておくことが有効であり、ROPでソートする。

### 3.3 副問合せ処理方式

限定述語を含む問合せの場合には、副問合せ側の表、主問合せ側の表の順で検索を行う。後者については、プリプロセス時または実行時に、限定述語の判定をCSPまたはCPUで行うかを判断する。以下に、DBMSが実現している副問合せ処理方式を示す。また、表1に副問合せ処理方式選択法を示す。

(1) 比較述語変換方式

以下のように比較述語に変換し（プリプロセス時）、副問合せ結果の複数の要素の値をその最大値、または最小値に置換し（実行時）、主問合せ側の表の条件に組み込み検索する。

- >ANY 副問合せ → >MIN
- >ALL 副問合せ → >MAX
- <ANY 副問合せ → <MAX
- <ALL 副問合せ → <MIN
- >=ANY 副問合せ → >=MIN
- >=ALL 副問合せ → >=MAX
- <=ANY 副問合せ → <=MAX
- <=ALL 副問合せ → <=MIN

ただし、MIN、MAXは副問合せ結果の要素の最小値、最大値を各々意味する。

(2) IN述語変換方式

限定述語をIN述語に変換し（プリプロセス時）、副問合せ結果の要素をIN述語に展開し（実行時）、主問合せ側の表の条件に組み込み検索する。（図2）

- =ANY 副問合せ → IN()
- <>ANY 副問合せ → NOT IN()

ただし、()の中は、副問合せ結果の要素値が列挙されていることを表す。

(3) ソート・マージ方式

- ① 限定述語を除いた検索条件で主問合せ側の表を検索し一時表T2に格納する。
- ② 副問合せの結果が格納されている一時表T1とT2を

ソートし、ソート結果をマージすることにより、限定述語を満足するT2の行を出力する。

(4) ネステッドループ方式

- ① 限定述語を除いた検索条件で主問合せ側の表を検索し一時表T2に格納する。
- ② 副問合せの結果が格納されている一時表T1とT2をネステッドループにより比較し、限定述語を満足するT2の行を出力する。

4. おわりに

限定述語についても動的最適化で実行時に効率の良いアクセスパス・演算方法を選択し、CSP・ROPを用いて処理することにより、CPUで副問合せ結果1件毎に比較を行う従来方式と比べ、大幅な性能向上を実現した。

[参考文献]

- [1] 井上他「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] 中村他「データベースプロセッサRINDAの最適化方式」、情報処理学会第37回全国大会5Q-8、1988
- [4] 速水他「データベースプロセッサ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「データベースプロセッサRINDAの関係演算方式」、情報処理学会第37回全国大会5Q-6、1988

表1 副問合せ処理方式選択法

比較演算子	限定子	判断条件	判断時期	副問合せ処理方式
>, <, =, <=	ANY, ALL		プリプロセス時	比較述語変換方式
=, <>	ANY	副問合せ結果をIN述語に展開できる	実行時	IN述語変換方式
=	ANY	副問合せ結果をIN述語に展開できない	実行時	ソート・マージ方式
上記以外			プリプロセス時	ネステッドループ方式

[検索命令]

```
SELECT *
FROM 会社表
WHERE 地区 = '東京' AND 社番 = ANY (SELECT 社番
FROM 部品表
WHERE 在庫 (<= 50))
```

[処理の流れ]

- ① 部品表の検索  
条件：在庫 <= 50
- ② 検索結果の行数の判定  
→ IN述語化
- ③ 会社表の検索  
条件：地区 = '東京' AND  
社番 IN (5, 3, 2, 10, ...)

[データの流れ]

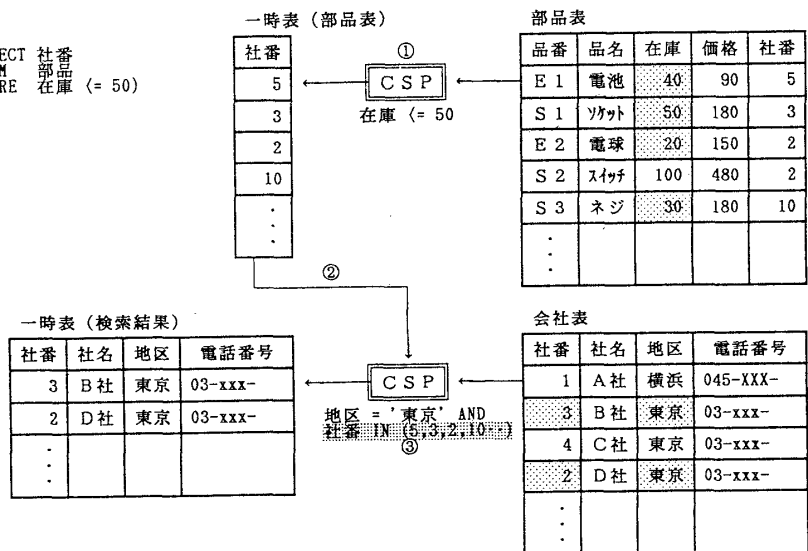


図2 RINDA使用による副問合せ処理方式例