5Q-9

データベースプロセッサＲＩＮＤＡのデータベースアクセス方式

板倉　一郎、　中村仁之輔、　井上　潮

ＮＴＴ情報通信処理研究所

１．はじめに

リレーショナルデータベース処理の性能向上を図るため、サーチ・ソートを高速に処理するデータベースプロセッサＲＩＮＤＡ[1]を開発した。本稿では、ＲＩＮＤＡを制御するＤＢＭＳのデータベースアクセス方式について述べる。

２．検索処理方式

ＲＩＮＤＡは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。

ＣＳＰはディスクに格納された表を指定された条件でサーチし、条件に合致する行に対し射影を行いＣＰＵに転送する。ＲＯＰはＣＰＵから転送された表の行のソート、ふるい落とし、重複除去等を行い、結果をＣＰＵに転送する。

ＣＳＰの出力とＲＯＰの入出力が複数行単位であるため、検索処理は一時表（検索の途中結果を格納する一時的な表でディスク／バッファ上に存在し、検索終了後削除される）を単位として、以下のような手順で実行される（図１）。

①ＣＳＰで表を検索し結果をＴ１（一時表を意味し、以下の説明でも同様）に格納する。

②Ｔ１に対しソフトウェアで処理する条件式があればその判別をＣＰＵで行い、結果をＴ２に格納する。

③ソート、結合、グループ化が指定されている場合は、Ｔ２をＲＯＰでソートし結果をＴ３に格納する。

④結合、副問合せが指定されている場合は、①～③をそれぞれの表に対して行い結果を一時表に格納し、ＣＰＵで結合、副問合せを行う。

なお、表をサーチしその結果をソートするという単純な問合せについては、ディスク／バッファ間のＩ／Ｏ時間を削減するため、以下の手順で処理するＣＳＰ／ＲＯＰ連動方式を実現している。

①ＣＳＰのサーチ結果をバッファに転送する。

②バッファのデータをＲＯＰに転送する。

③ＣＳＰのサーチを再開し、①、②を繰り返す。

④ＲＯＰからのソート結果を一時表に格納する。

問合せがＣＳＰ／ＲＯＰ連動で処理可能であるかは言語処理部で検索条件式の解析により判断する。

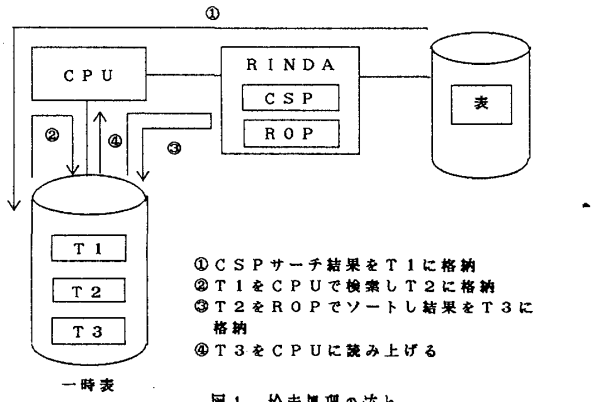

図１　検索処理の流れ

３．バッファ管理方式

３．１　要求条件

ＣＳＰ・ＲＯＰを有効に利用し、問合せを処理するため以下の処理が効率良く実行されなければならない。

①ＣＳＰのサーチ・ＲＯＰのソートのためのデータ転送

②一時表への連続ページアクセス

一般的に使用されているページ単位のバッファ管理方式では、次の問題があり効率良く検索を実行することができない。

①ＣＳＰ・ＲＯＰの起動がページ単位となり、ＣＳＰ・ＲＯＰ起動回数が多くなる。

②処理の単位が一時表であるため、連続的にページをアクセスすればよいにもかかわらず、ページ単位の入出力ではディスクの　シーク・サーチ動作が多くなる。

３．２　ＲＩＮＤＡ用バッファ管理方式

(1) 実現機能

上記の問題点を解決するため、ＲＩＮＤＡ用のバッファ管理として以下の機能を実現した。

①ＣＳＰ／ＲＯＰ起動回数を減らすため、バッファを複数ページ単位（トラック単位）で一時表に割当てる。

②一時表のページは基本的にはシーケンシャルにアクセスされることを考慮し、シーク・サーチ動作回数を削減するため、バッファ／ディスク間の入出力をトラック単位で行う。

また、バッファを有効に利用するため、次の機能も実現した。

③一時表にアクセスする時点でバッファを確保し割当てる。ただし、結合処理時には３個の一時表（入力２個、出力１個）が必要となることを考慮し、最大３個までバッファを動的に確保する。

(2) 動作概要

(1)の機能を実現したバッファ管理方式の動作概要を以下に示す。

(A) 一時表アクセス時のバッファの割り当て

３個の一時表まで動的にバッファを確保し割当てる。

ＲＩＮＤＡ－Relational Database Processor: Database Access Method

Ichirou ITAKURA, Jinnosuke NAKAMURA, Ushio INOUE

NTT Communications and Information Processing Laboratories

３個使用中であればＦＩＦＯによりバッファを割当てる。

(B) ページアクセス時のバッファの割当て

バッファ上にアクセスするページが存在するか調べる。存在しないならば、アクセスするページを含むトラック全体をバッファに読み込む。ただし、既にバッファ上に追い出す必要のあるページが存在する場合には、トラック単位でディスクに書き込む。

(C) 動作例

一時表Ｔ１のP1～P15を５ページ単位でＲＯＰに転送し、P6'～P10'をバッファに転送する直前のバッファの状態を示す（図２）。

4　ディスク並列アクセス方式

ＲＩＮＤＡを用いた検索処理では表全体のサーチを行うため、表のデータ量に比例した処理時間がかかる。表のサーチ時間を短縮するため、表を複数のボリュームに分散格納し、複数台のＣＳＰで並列にサーチするディスク並列アクセスを実現した（図３）。

４．１　分散格納

表の複数ディスクへの分散格納実現に当り以下の事項を考慮した。

①各ディスクに均等にデータを格納する。

②格納ディスク決定のためのオーバヘッドが少ない。

特定の列の値に基づきハッシング、クラスタリング等により格納ページ・格納ディスクを決定する方法は、列の値の分布による偏り、列の値の更新に伴う行の移動が発生する可能性があり、必ずしも良い方法とは言えない。このため、行は到着順に格納し、ページ単位で周期的にディスクに割当てる方法を実現した。

４．２　並列検索

ディスク並列アクセス時には以下のようにバッファを使用し、並列に検索を実行する。

①一時表に割当てられたバッファを並列にサーチを行うＣＳＰに均等に割当てる。

②複数台のＣＳＰを起動し、並列に検索を行う。

③ＣＳＰの検索動作は、指定された検索範囲のサーチが終了、あるいは、検索範囲のサーチは終了していないが検索結果を格納するバッファが満杯になったときに終了する。前者で終了し未使用バッファが存在する場合は、バッファ不足で検索が中断しているＣＳＰにバッファを再度割当て、検索を再開する。

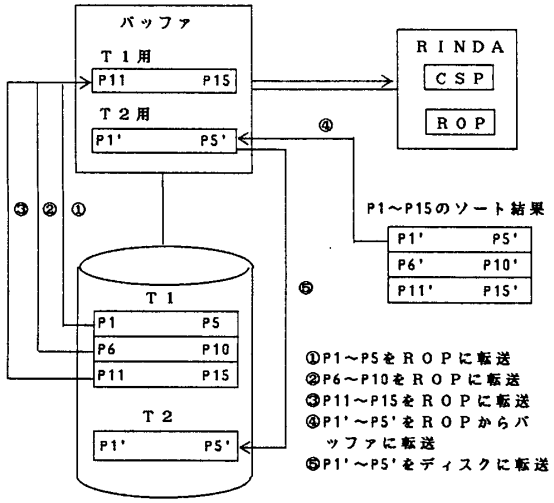

図２　バッファ管理の動作例

5．おわりに

本稿では、ＲＩＮＤＡのデータベースアクセス方式として検索処理、バッファ管理、ディスク並列アクセスについて述べた。一時表を単位とした処理方式に着目したバッファ管理方式により、一時表を格納するディスクとのＩ／Ｏ時間短縮を図った。

6．参考文献

〔1〕井上他「データベースプロセッサＲＩＮＤＡのアーキテクチャ」、情処第37回全国大会

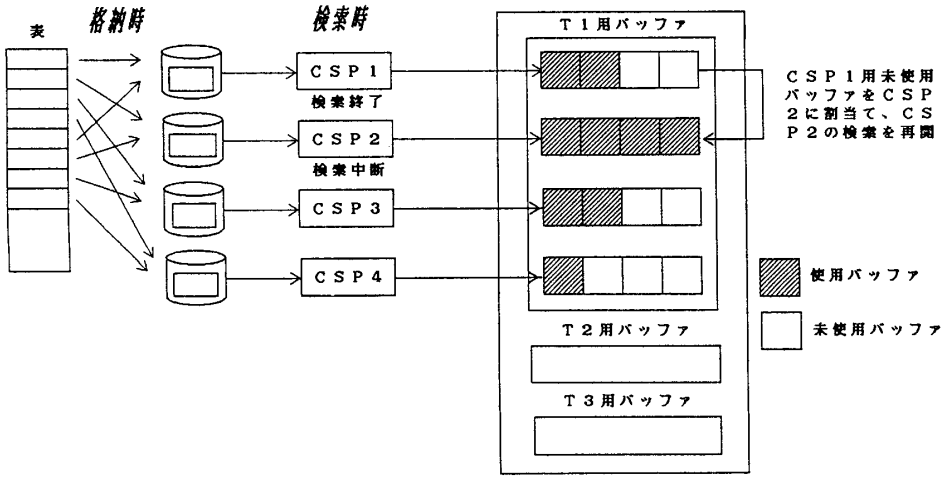

図３　ディスク並列アクセス方式の概念