

## 複合検索のための複合オブジェクト索引の並列処理方式

5 A A - 4

小倉 一泰 都司 達夫 宝珍 輝尚 (福井大学工学部)

## 1. 前書き

我々は複合オブジェクトに付与された索引を水平垂直に分割し、分割した索引を独立した PE に格納し、これらの PE を用いて複合オブジェクトに対する検索処理を並列に行う方式を提案した[2]。本研究ではこの方式を複合検索が可能のように拡張した。この拡張では集合を属性とするオブジェクトを含むパス式に対する効率的な検索も同じように可能とすることができる。以下では基本となる並列検索方式の概要を説明した後この拡張について述べる。

## 2. 並列検索方式

本研究では複合オブジェクトの索引方式として並列化への適応性の点からマルチインデキシング方式[1]を採用している。マルチインデキシングの各サブインデックスを垂直分割と呼び、垂直分割をさらに水平に分割し、この各々を水平分割と呼ぶ。

水平垂直方向に分割した各サブインデックスを独立した PE に割り当て、各 PE が協調して動作する事で検索処理を行う。

## 3. 複合検索と集合の検索処理

複合検索とは一つのオブジェクトに対し複数の検索条件を指定し個々の検索条件を満たす検索結果集合を集合演算することにより検索結果を得る検索のことである。A を検索のターゲットクラスとして、B, C, D, E, F を複合するクラスの属性とする。

$$A.B.C.D = "abc" \text{ and } A.E.F = "def"$$

この複合検索式は集合積に属さない個々の検索結果に対してパス式 A.B において余分に検索することをさけるために以下のように正規化してから検索する。

$$A.B = (B.C.D = "abc" \text{ and } B.E.F = "def")$$

本研究ではこのように正規化した複合検索を取り扱う。

このような形の複合検索を実行する場合  $B.C.D = "abc"$ 、 $B.E.F = "def"$  それぞれの検索結果を得た後並列検索システムの外に取り出し、集合積

を計算し、その結果を新たな検索要求としてパス式 A.B を検索する方法が考えられる。

この方式の問題点としては

1. 途中の検索結果を並列検索システム外に取り出すオーバーヘッド
2. 集合演算の処理において検索システムの並列性を利用できないこと
3. 集合演算結果を新たな検索要求として並列検索システム中の適切な PE (Processor Element) に再配布するオーバーヘッドがあげられる。

また、集合オブジェクトを含むパス値とは図 1 のようなパス値のことである。パス式中の属性が集合を持ったときの問題点はこのようなパス値を検索したときに結果集合に集合オブジェクトが複数格納されることであり、もしそれが検索の最終段ではなく検索パスの途中に現れたときにはそのオブジェクトを参照するオブジェクトを複数回検索してしまうことによる無駄な処理の発生が問題となる。

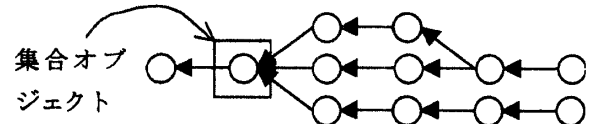
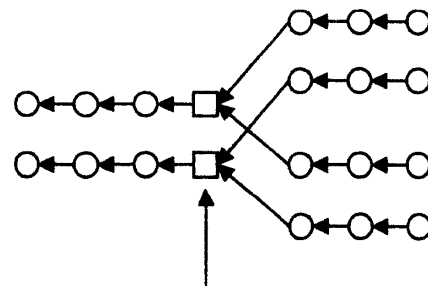


図 1 集合オブジェクトを含むパス値

## 4. 集合演算の処理

前節の問題点を解決するために本研究では集合演算を検索システム内で並列に行うことを考える。

集合の操作の並列化について考えると集合はその要素ごとに独立に演算を行うことができ、集合とし



結合バッファ

図 2 複合検索における集合演算の並列処理

での演算結果は要素ごとの演算結果の集合となる。本研究が前提とした並列検索システムではオブジェクトはただ1つのPEにのみ存在することが保証されている。従って集合演算は各オブジェクトを管理する複数のPEで並列に行うことができる(図2)。さらに基本とした並列検索方式では各PEは検索結果がすべて求まることを待つことなく次のPEに送る。これにより検索処理を各PEでオーバーラップさせることでパイプライン並列化を実現している。このため、集合演算処理も集合の要素すべての到着を待つことなく演算を行い、各PEを効率的に動作させる必要がある。

ここでは、集合演算処理は到着したオブジェクトを登録した辞書を用いて行う。この辞書を結合バッファと呼ぶ。結合バッファの使用方法は集合演算毎に異なるため、個々に次で述べる。

#### 4.1. 和集合演算の処理

結合バッファを到着済みの検索結果を格納するバッファとして用いる。つまり、検索結果が到着するごとに、結合バッファを検索し、登録されていない検索結果のみを検索するために結合バッファに登録する。

#### 4.2. 積集合演算の処理

結合バッファを検索結果一つ一つに対していくつかの検索条件を満足しているかのカウントを保持する辞書として用いる。つまり、 $n$ 項演算のAND( $n$ 項がすべてandで結合されている演算式)では検索結果が到着する度に結合バッファを検索し、結果それぞれのカウントを得る。結合バッファに登録されていなければ新たにカウントを1として登録する。登録されていればそのカウントに1加えてカウントを更新する。更新の結果 $n$ に達した場合はand条件をすべて満たしているのでそのオブジェクトに対する検索を行い、そのオブジェクトは結合バッファから削除する。

#### 4.3. 否定集合演算の処理

結合バッファを未到着バッファとして用いる。結合バッファはそのPEが管理するすべてキーを登録して初期化される。検索結果が到着する度に結合バッファから到着した結果を削除する。検索が終了したとき、結合バッファに残っている要素がNOTの結果である。これを次のPEに送信する。

#### 4.4. 集合属性をもつパス式の処理

集合属性をもつパス式を検索するときには、3で

述べた問題点が発生する。これを防ぐためには4.1と同じような処理を施す。つまり、集合オブジェクトを検索したときには結合バッファに登録し、一度検索した集合オブジェクトは再度検索しないようにする。

### 5. 終了判定

4で述べた集合演算が終了したかの判定を行う必要がある。この理由は以下の通りである。

- 1 結合バッファ破棄のタイミングを与える。
- 2 NOT 結合処理において結合処理以前の検索の終了を検出し、NOT 結合処理の終了を判定し、下流の検索行程に結合処理の結果を検索要求として送出する。
- 3 複数の結果をパケットにまとめ、次段のPEに送るためのパケットのフラッシュのタイミングを与える。

本研究では終了判定手法については文献[3]で提案した終了判定手法を上記の目的に沿うよう拡張して使用しているが、詳細は別稿にゆずる。

### 6. 結論

我々の提案した並列検索システムの枠組みの中で集合オブジェクトを含むパスと複合検索を効率よく並列処理する方式を提案した。今後は効率的な終了判定の方法を模索し本方式を実装する事により本方式の有効性を実証したい。なお、本研究は平成9年度文部省科学研究費重点領域研究「高度データベース」(課題番号09230207)の援助を受けている。

### 7. 参考文献

- [1] E.Bertino, "A Survey of Indexing Techniques for Object-Oriented Database Management Systems", in *Query Processing For Advanced Database Systems*, Freytag, J.C., Maier, D. and Vossen, G. (Eds.), pp. 383-418, Morgan Kaufmann, 1994.
- [2] 小倉、都司、Vreto、宝珍、"複合オブジェクトの索引に対する水平垂直分割の一方式", 信学論VOL.J80-D-I NO.6 pp486-494
- [3] K.Ogura, T.Tsuji, T.Hochin, "Index splitting for Complex Object in Parallel Environment", *Proceedings of International Symposium on Cooperative Database Systems for Advanced Applications*, 1, 19-22(1996.12).