

3M-04 複合オブジェクトに対する並列複合検索システム*

○福岡 久長† 樋口 健 都司 達夫 宝珍 輝尚 (福井大学大学院工学研究科‡)

1 はじめに

他のオブジェクトを構成要素として参照しているオブジェクトは、複合オブジェクトとよばれる。また、オブジェクトは複数の上位オブジェクトから参照されている場合がある。検索システムではその参照関係に索引を付けて検索が行なわれるが、並列検索システムでは、並列計算機の各 PE(Processor Element) に索引を分割し、それを並列に検索することにより検索の高速化をはかる。

本研究では複合オブジェクトの索引方式としてマルチインデキシング方式を使っている。

2 Cenju-3

本研究で使われている並列計算機は、NEC の Cenju-3 である。Cenju-3 は分散メモリ型並列コンピュータであり、最大 256 個の PE が多段接続網によって接続されている。各 PE は 150MIPS, 50MFLOPS の性能を持ち、各 PE との通信に 40MB/sec の通信能力を持っている。

3 複合オブジェクトの検索

複合オブジェクトのパスは $P = C_1 A_1 A_2 \cdots A_n$ と表すことができる。複合オブジェクトの検索とは、このパス式の参照関係を逆順にたどり、あるオブジェクトの最終的な参照元オブジェクトを特定することである。

また、パス P とオブジェクトの集合 S に対して $R(P, S)$ をパス P 上での集合 S の要素であるオブジェクトを参照するクラス C のインスタンスの集合とする。

4 複合検索

複合検索とは、AND や OR などの論理式で結合された複数の検索条件を検索することである。複合検索の検索条件は、パス P_1, P_2 、さらに共有パス P_3 に対して、" $P_1 = a$ " opr " $P_2 = b$ " sharing P_3 と表される。

ここで P_3 とは、 P_1 と P_2 の共通部分の接頭辞であり、opr は "and" または、"or" である。そして、その検索結果は $R(P_3, R(C'P'_1, \{a\}) \text{ opr } R(C'P'_2, \{b\}))$ である。

ただし、 $P_1 = P_3 P'_1$, $P_2 = P_3 P'_2$, C' は P_3 の定義域である。つまり AND 検索の場合は、" $P_1 = a$ " and " $P_2 = b$ " sharing P_3 となり、検索結果は、 $R(P_3, R(C'P'_1, \{a\}) \cap R(C'P'_2, \{b\}))$ である。同様に、OR 検索の場合は " $P_1 = a$ " or " $P_2 = b$ " sharing P_3 となり、検索結果は $R(P_3, R(C'P'_1, \{a\}) \cup R(C'P'_2, \{b\}))$ である。

例えば $P_1 = CA_1 A_2 A_3 A_4$, $P_2 = CA_1 A_2 B_3 B_4$, $P_3 = CA_1 A_2$ の複合検索の場合、 $P'_1 = A_3 A_4$, $P'_2 = B_3 B_4$ となる。そして P'_1, P'_2 で検索した後にその結果に対し AND 検索の場合は AND の集合演算処理を行ない、OR 検索の場合は OR の集合演算処理を行なう。その後、その検索結果を共有パス P_3 で検索する。

5 結合バッファ

複合検索では、共有パスに関する部分の検索を行なう直前に、AND や OR などの集合演算処理を行なう必要がある。この集合演算処理を、到着したオブジェクトを登録するための結合バッファを用いて行なう。検索において論理演算の生じるクラス (Cell) に集合演

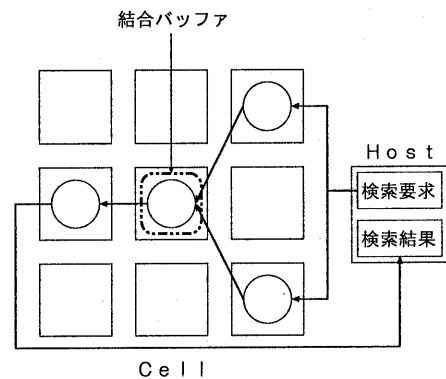


図 1: 結合バッファ

算のための結合バッファを用意する。結合バッファの使用法は、AND、OR といった演算毎に異なるので、それぞれについて使用法を述べる。

5.1 AND 演算での動作

AND 演算では、集合演算の生じるクラスにおいて、もう一つのパスから送られてきた検索結果の参照元のオブジェクトが結合バッファに登録されているかを調べる。そこで結合バッファに登録されているならば、

*Parallel Complex Retrieval System for Complex Object Index

†Hisanaga Fukuoka

‡Graduate School of Engineering, Fukui University

AND の条件を満たしているので次の共有部分の検索を行なう。結合バッファに登録されていないときは、結合バッファに登録するだけである。

5.2 OR 演算での動作

OR 演算でも、まず集合演算の生じるクラスにおいて、送られてきた検索結果の参照元のオブジェクトが結合バッファに登録されているか調べる。登録されていないならばオブジェクトを登録し、次の共有部分の検索を行なう。登録されていれば、同じ検索結果を検索要求として2重に検索することになるので検索しない。

6 終了判定機構

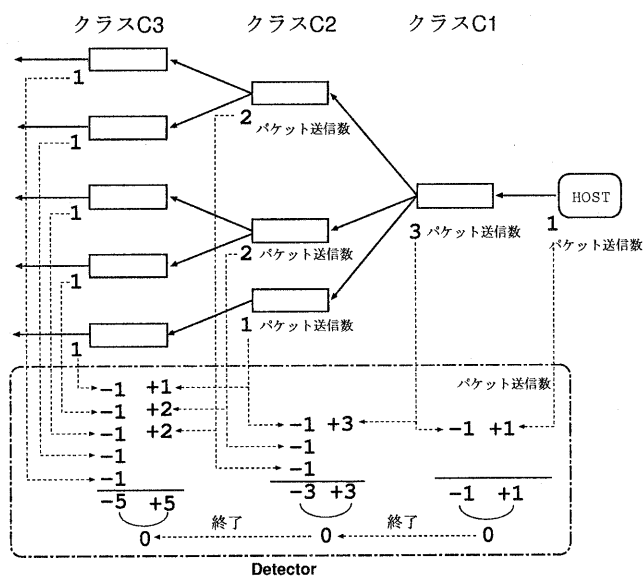


図 2: 演算終了判定機構

多くの複合検索要求がストリームとして連続的に送られてくると、結合バッファ内のデータサイズは非常に大きなものとなってくる。よって集合演算が終了し、使用されなくなった結合バッファ内の登録データは削除する必要がある。また、索引の更新・削除において検索がどこまで終了したかをしる必要がある。そこで、検索終了判定機構では現在の検索がどこまで終了したかを特定する Detector と呼ばれるものを使いバッファの削除・索引の更新などのタイミングを計る。Detector の流れを図 2 に示す。

Detector では図 2 のように各段 (クラス) に対応するカウンタを持ち、検索要求のためのパケットを次の段への検索に送った数を Detector に送り、Detector でその数を段ごとにカウンタでプラスする。また各 PE において、送られて来たパケットの検索が終了したら、Detector に終了したことを知らせる。和演算では送られてきた検索結果の参照元のオブジェクトが結合バッファにあれば検索を Detector はそれを受け取ったら、

PE で検索終了した段に対応するカウンタを 1 減らす。その段でのカウンタの数字が 0 になり、かつ前の段での検索が終了していたらその段での検索の終了である。その時点で、その段での検索に使った結合バッファを削除することが出来る。最終的に Detector の中のすべての段のカウンタの数字が 0 になったら検索の終了である。

7 更新・削除について

並列検索システムでは、検索要求、更新、削除の要求は順に Host に与えられる。

更新要求を処理するためには、それまでに出された検索要求が確実に終了されてなければならない。また、更新要求が処理されないうちは、更新要求の以後の検索要求は実行できない。

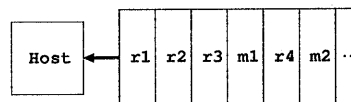


図 3: 検索、更新要求の例

図 3 は要求の例である。r は検索の要求、m は更新の要求である。検索要求 r1~r3 は更新要求 m1 の前に行われなければならない。そして、検索要求 r4 は m1 の後、m2 は r4 の後に行われるようにしなければならない。

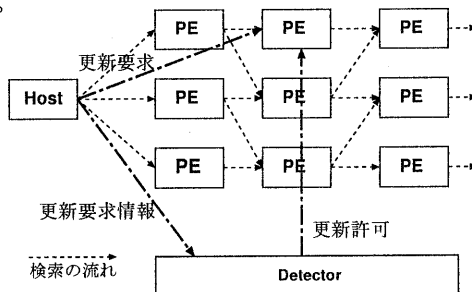


図 4: 更新の例

並列検索システムでは、Host から送信された要求の順序が、そのままある PE に届く保証が無い。検索要求は、順に索引が存在する PE に検索要求を送って行くが、更新要求では、その索引が存在する PE に直接送り付ける。こうすることによって、更新要求を、それ以後の検索要求より先に到着させることが出来る。また、ある PE において、r3 が m1 よりも後に要求が到着する可能性がある。各 PE において、現在どれだけの検索要求が終了したかを知り、先に届いた更新要求を処理して良いか知る必要がある。Detector によって、検索の各段 (クラス) での検索終了を知ることが出来るので、更新要求以前の検索要求が、その段において終了したことが分かれば、更新の許可を PE に送り、更新要求を処理できる。