2R−7

複合オブジェクトの集合演算のための索引構造

岩井原 瑞穂　牧之内 顕文
(九州大学 工学部 情報工学科)

1 まえがき

複合オブジェクトとは，複数のオブジェクトを集合型やリスト型などに集約して1つのオブジェクト(集約オブジェクト)とした，複雑な構造を持つデータである．集約オブジェクトに対する検索として，与えられた集合との定められた包含関係を満たす集合を持つオブジェクトを求めるものがある．このような検索を効率的に処理するデータ構造として，もし集合の要素の値の取る範囲が大きくなければ，各集合ごとにそれが含む要素を表わすビットベクトルを用意するのが有効である．しかし集合の要素の値の取る範囲が大きいときは，索引やシグネチャ[2]を用いるのが有利である．

本稿では，要素をキーとする索引を各集約オブジェクトごとにそれぞれ用意する局所索引と，要素の全体集合の索引を1つ用意し，各要素についてそれを含む集約オブジェクトを求めることのできる全体索引の2つについて，性能比較を行なった．局所索引はいくつかの商用オブジェクト指向データベースに見られる方法である．全体索引は文献[1]で提案されている索引と基本的に同じであるが，文献[1]では集約オブジェクトでの包含関係による検索は想定されていない．本稿では，与えられた集合を包含する集約オブジェクトの検索について，上記2種類の索引の処理時間を解析した．

2 集合演算に基づく複合オブジェクト検索

複合オブジェクトとは，複数のオブジェクトを集約して1つのオブジェクトとする，入れ子構造を持つオブジェクトである．集約の型として集合，重複集合，リスト，配列などがあるが，これらをまとめて**集約オブジェクト**と呼ぶことにする．集約オブジェクトに対する検索として，その要素集合に対する集合演算で定義されるものがある．集約オブジェクトは様々な集約の型を持っており，直接集合演算が定義できないため，これを集合型に変換したものを**パタン**と呼び，このパタン上で集合演算を定義する．要素の集合が**質問パタン**として与えられ，このパタンとデータベース側の集約オブジェクトのパタンとの包含関係により，次のような条件を満たす集約オブジェクトの検索が記述できる．

(包含パタン) 質問パタンを包含するパタンの集約オブジェクト
(被包含パタン) 質問パタンに包含されるパタンの集約オブジェクト
(一致パタン) 質問パタンと一致するパタンの集約オブジェクト

例として，次のオブジェクト識別子(OID)，車種名(cartype)，その色の集合(colors)からなるオブジェクトの集合(クラス)を考える．ここでcolorsは集合型の集約オブジェクトである．

OID: s1, cartype: sedan, colors: { gray, white }
OID: s2, cartype: roadstar, colors: { red, silver, yellow, white }
OID: s3, cartype: minivan, colors: { gray, red, white }
OID: s4, cartype: convertible, colors: { silver, red, yellow }

質問パタンとして，Q = {gray, white}が与えられたとき，それを包含するcolorsを持つオブジェクトは$s1$と$s3$である．同じQに対し，それに包含されるcolorsを持つオブジェクトは$s1$である．

以下，集約オブジェクトを持つオブジェクトを親オブジェクトといい，そのクラスを$\mathcal{S}$とする．集約オブジェクトの要素オブジェクトのクラスを$\mathcal{N}$とする．親オブジェクトのOIDを略してSIDと呼ぶ．

3 集約オブジェクト検索のための2つの索引構造

本節では，前節で示した複合オブジェクト検索に適した二次記憶の索引構造を2つ示す(図1)．

3.1 局所索引

各親オブジェクト$s_j \in \mathcal{S}$について，その要素オブジェクトのoidまたは値で構築する索引を，**局所索引** (local index; LIX)とする．つまり親オブジェクトの数$|\mathcal{S}|$と等しい数の索引が作られる．また親オブジェクトのうち，集約オブジェクト以外の属性値は組s'_jとして1つのクラスタにまとめられる．各組s'_jにはその局所索引の要素数を保持しておく．

3.2 全体索引

クラス$\mathcal{N}$のオブジェクトのOIDまたは値をキーとして構築した索引を**全体索引** (global index; GIX)と呼ぶ．各キー値ごとに，それを集約オブジェクトの要素として持つ親オブジェクトのSIDのリスト(**SIDリスト**)へのポインタを置く．GIXは要素オブジェクトから親オブジェクトへの逆引き索引と考えられる．

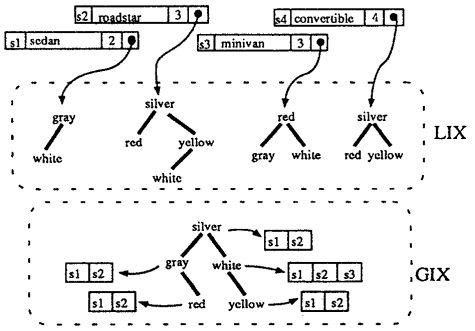

図1: 局所索引(LIX)と全体索引(GIX)

4 包含パタンを求めるアルゴリズム

集約オブジェクトの検索のうち，包含パタンを求める検索について，全体索引を用いるアルゴリズムと局所索引を用いるアルゴリズムの2種類を紹介する．

4.1 全体索引を用いた検索アルゴリズム − GIX-SUP

(ga1) 質問パタンQを読み込み，GIXをアクセスするIOコストを減らすため，Qをその値でソートする．

(ga2) ソートされた順にQの値を取り，それでGIXをアクセスし，SIDリストを得る．SIDリストが1つ得られるごとに(ga3)を実行する．

(ga3) (ga2)で得られるSIDリストのSIDの共通集合を求めれば，それが答になる．その共通集合は，最初に得られたSIDリストのSIDをキー値として主記憶内にハッシュテーブルを作り，各SIDが残りのSIDリストに何回現われたかハッシュテーブルを用いて数えることによって求める．全てのSIDリストに含まれていたSIDを答えとして出力する．もし途中で答となり得るSIDがなくなれば，「答なし」と出力し，終了する．

Efficient Indices for Set Operations of Complex Objects
Mizuho Iwaihara and Akifumi Makinouchi
Dept. Computer Science & Comm. Eng., Kyushu University

4.2 局所索引を用いた検索アルゴリズム − LIX-SUP

(la1) 質問パタン Q を読み込み，ソートする

(la2) 親オブジェクトの組 s'_j を順に読み込み，その集約オブジェクトの要素数が $|Q|$ より小さくないものを選択する．

(la3) (la2) で選択された s'_j が1つ得られるごとに，s'_j の LIX をアクセスし，Q の値が全て LIX にあるか調べる．途中で1つでも LIX になければ，次の s'_j に移る．全てあれば s'_j の SID を出力する．

5 処理時間の解析

5.1 集約オブジェクトの分布モデル

前節の包含パタン検索のための2つのアルゴリズムについて，それらの実行時間の期待値を理論的に求め比較する．そのために，データベースの集約オブジェクトの分布について，次のモデルを用いる．1つの要素オブジェクトが1つの集約オブジェクトの要素となる確率は，オブジェクトの選び方によらず一定の数 E と仮定する．そしてこの E と集約オブジェクトの数 S，および要素オブジェクトの数 N をパラメータとして変化させてみる．上記の分布モデルにおいて，1つの集約オブジェクト s_j が，k 個の異なる要素オブジェクトを持つ確率を考えてみる．これは，k 個の要素オブジェクトが s_j の要素となり，残りの $(N-k)$ 個が要素とならない確率 $E^k(1-E)^{N-k}$ に，k 個の要素の選び方の数 $\binom{N}{k}$ を掛けた下式となる．

$$b_N(k) = \binom{N}{k} E^k (1-E)^{N-k}.$$

これより，集約オブジェクトの個数は，その要素数 k の二項分布となることが分る．その平均は $N \cdot E$，分散は $N \cdot E(1-E)$ である．

5.2 解析結果とその解釈

前節のモデルに基づいて，2つのアルゴリズム GIX-SUP と LIX-SUP のコスト計算式を導出したのであるが，紙面の都合上詳細な解析式は割愛し，代わりに図2の数値例をもとにアルゴリズムの振舞いを説明する．解析に用いたパラメータを表1に示す．ここでは局所索引と全体索引ともに B 木を用いている．

検索において，答が1つ以上見つかる場合 (成功検索) と，1つも答がない場合 (失敗検索) に分けられる．アルゴリズム GIX-SUP では，成功検索と失敗検索とでは処理時間が大きく異なると予想されたため，成功検索時間の解析と，成功検索と失敗検索の時間をそれらの起る確率を掛けて加えた 合併検索時間 の解析の両方を行った．しかし LIX-SUP では両者に大きな差がなかったため，合併検索時間のみグラフに示してある．

まず，アルゴリズム GIX-SUP の振舞いについて説明する．(ga1) の IO とソートの合計時間への寄与は小さい．GIX-SUP では成功検索のとき，質問パタンの要素数 Q と同じ回数 GIX がアクセスされ，同数の SID リストが読みこまれる．この IO コストが GIX-SUP の成功検索時間の大部分を占め，グラフ (破線) は SID リストの個数に比例した直線になる．1つの SID リスト中の SID の平均個数は $S \cdot E$ である．

包含パタンの検索では，答の個数の期待値は $S \cdot E^Q$ で与えられる．このため Q の増加とともに失敗探索の可能性が大きくなる．数値例では数10回 GIX をアクセスし，SID リストを読み込んだ段階で失敗検索と分り GIX-SUP が終了する可能性が非常に高くなる．このため，GIX-SUP の併合探索時間 (一点鎖線) は，数10回の GIX アクセスと SID リスト読み込みでほぼ平坦なグラフとなる．以上より，GIX-SUP は合併検索時間では小さいが，Q が大きいときに探索がまれに成功すると非常に大きな処理時間がかかることが分る．

パラメータ	定義	値
S	集約オブジェクトの数	10000
N	要素オブジェクトの数	10000
E	ある要素オブジェクトがある集約オブジェクトの要素になる確率	0.1, 0.5, 0.9
Q	質問パタン Q の要素数	$1 \leq Q \leq N$
P	1ページの大きさ	4096 バイト
IO	1ページを読むまたは書く時間	20 ミリ秒
b	B 木の平均ファンアウト数	200
t_S	親オブジェクトの組 s'_j の大きさ	100 バイト
t_G	SID の大きさ	8 バイト
t_Q	要素オブジェクトのキー値の大きさ	8 バイト
$comp$	2つの値を比較する CPU 時間	0.2 マイクロ秒
$move$	1つの組を移動する CPU 時間	0.1 マイクロ秒
$hash$	1つのハッシュ値計算の CPU 時間	0.6 マイクロ秒
F	ハッシュ検索における平均探索回数	1.4

表 1: 解析に用いたパラメータ

次に LIX-SUP の振舞い (実線) について説明する．(la2) で要素数に対する条件 ($\geq Q$) を満たす集約オブジェクトが選択され，その LIX へのみアクセスが行なわれるが，Q が要素数の分布の平均 $N \cdot E$ より小さい段階では，選択率が大きく，ほとんどの LIX へアクセスが行われる．また Q が要素数の平均を越える辺り ($E = 0.1$ のとき 1000，$E = 0.5$ のとき 5000，$E = 0.9$ のとき 9000) から，(la2) での選択率が小さくなり，LIX のアクセスコストが小さくなる．そして Q が十分に大きいときのコストはほとんど (la1) と (la2) だけになる．

以上より，GIX-SUP と LIX-SUP を比較すると，Q の小さい領域では GIX-SUP の方が効率が良く，Q が集約オブジェクトの要素数の平均を越える領域では，要素数による選択を行なう LIX-SUP の方が効率が良いといえる．

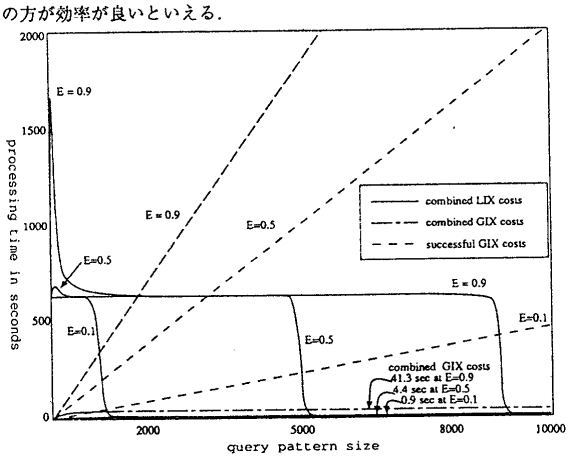

図 2: アルゴリズム GIX-SUP と LIX-SUP の実行時間

6 まとめ

集約オブジェクトに対する包含パタンの検索で，局所索引と全体索引の2種類を用いたアルゴリズムの性能比較を行なった．今回用いたパラメータでは全体索引の方が効率が良かったが，パラメータによっては効率が逆転することも考えられる．また局所索引はシグネチャと組み合わせてさらに効率化することが考えられる．

参考文献

[1] E. Bertino and W. Kim, "Indexing Techniques for Queries on Nested Objects," *IEEE Trans. Knowledge and Data Eng.*, 1(2), pp. 196-214, 1989.

[2] 福島, 石川, 干, 北川, 大保, "複合オブジェクトに対する索引機構の研究", 情報処理学会第 44 回全国大会, 2E-1, 1992.