3F－5

集合値検索を対象とした分割シグネチャファイル構成法の検討

渡辺 悟康 *　北川 博之 †

* 筑波大学工学研究科　† 筑波大学電子・情報工学系

1 はじめに

シグネチャファイルはテキスト検索を対象とした索引機構として提案され、レコード検索や集合値検索 [1] への応用が研究されている。一方、シグネチャファイルの構成法も *Sequential Signature File*(SSF)、*Bit-Sliced Signature File* (BSSF)、*Quick Filter*[2] 等が提案されている。

本稿では、シグネチャファイルによる検索コストの低減を目的とし、Quick Filter と BSSF を融合した新しいシグネチャファイル構成法である *Bit-Sliced Quick Filter* (BSQF) を提案する。また、集合値検索を対象として BSQF による検索コスト、及び格納コストについて見積りを行ない、その有効性を検討する。

2 シグネチャファイルを用いた集合値検索

シグネチャ (*signature*) とは、個々のデータオブジェクトから生成される固定長のビット列のことである。

シグネチャの作成法

1. 集合の各要素から、長さが F ビットで、その内 m ビットが "1" にセットされている**要素シグネチャ** (*element signature*) を作成する。(このセットされたビット数をウエイトと呼ぶ。)
2. すべての要素シグネチャのビットごとの論理和をとる**スーパーインポーズドコーディング** (*superimposed coding*) を行ない、**集合シグネチャ** (*set signature*) を作成する。

このようにして生成されたシグネチャと、各データオブジェクトの識別子 (OID) の組を格納したのが**シグネチャファイル** (*signature file*) である。

問い合わせが与えられた際に、問い合わせ条件中に現れる集合を**問い合わせ集合** (*query set*, Q)、データベース中の集合を**ターゲット集合** (*target set*, T) と呼ぶ。また、それぞれから作成される集合シグネチャを、**問い合わせシグネチャ** (*query signature*, S_Q)、**ターゲットシグネチャ** (*target signature*, S_T) と呼ぶ。

集合値検索の問い合わせ条件には様々なものがあるが、

記号	定義
N	オブジェクトの総数
D_t	ターゲット集合の要素数
D_q	問い合わせ集合の要素数
F, f	シグネチャのビット長
m	要素シグネチャのウエイト (= 2)
h	Quick Filter 部のキーのビット長
n	Quick Filter 部によって分割される partition の数 ($2^{h-1} < n \leq 2^h$)
A	アクチュアルドロップ数
Fd	フォルスドロップ確率
P_o	1 オブジェクトを取り出すためのページアクセス数 (= 1)

表 1: 変数の定義

本稿では、Q が T に含まれる場合 ($T \supseteq Q$) を対象とする。この問い合わせ条件の場合、$S_Q \wedge S_T \equiv S_Q$ を満たすターゲットシグネチャが、問い合わせ条件を満たす候補となる。しかし、これらの候補の中には、実際に問い合わせ条件を満たす**アクチュアルドロップ** (*actual drop*) と、実際には条件を満たさない**フォルスドロップ** (*false drop*) があるので、実際のデータからそれらを区別する必要がある。

3 Bit-Sliced Quick Filter (BSQF)

BSQF の概略を図 1に、用いる記号の定義を表 1に示す。

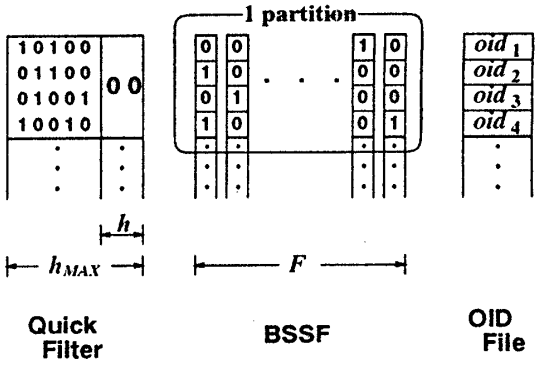

図 1: BSQF の構成

Evaluation of Partitioned Signature Files for Set-valued Object Retrieval

Noriyasu Watanabe* and Hiroyuki Kitagawa†

*Doctoral Degree Program in Engineering, Univ. of Tsukuba

†Institute of Information Sciences and Electronics, Univ. of Tsukuba

BSQF の作成法

1. 長さ F ビット、ウエイト m ビットでターゲットシグネチャを作成し、BSSF 部に格納する。
2. 上とは別のハッシングを用いて、長さを f ビット、ウエイトをフォルスドロップ確率が最低となる m_{opt} $(= \frac{f \ln 2}{D_t})$ ビットのターゲットシグネチャを作成し、その内 h_{MAX} ビットを Quick Filter 部に格納する。
3. 対応する OID を OID ファイルに格納する。

BSQF を用いた集合値検索では、まず Quick Filter 部で partition の絞り込みを行なう。続けて BSSF 部で問い合わせを満たす候補を取り出し、最後に実際のデータにアクセスし、アクチュアルドロップのみを返す。

4 コスト解析

まず仮定として、$n = 2^h$ であるとする。また、特に式を示さなかったものについては [1] の式を用いる。

4.1 検索コスト

総検索コスト RC は、Quick Filter 部で絞り込まれた partition 中の BSSF 部の検索コスト、OID ファイルにアクセスするコスト (LC_{OID})、候補となったオブジェクトがアクチュアルドロップかフォルスドロップかを実際のデータから区別するコストの和になるので、式 (1) で表される。

$$RC = n\mathrm{PAR}(h)LC_{BSSF} + LC_{OID} + P_o(A + Fd(N - A)) \quad (1)$$

ここで m_q を問い合わせシグネチャのウエイトとすると、$LC_{BSSF} = m_q$、また $\mathrm{PAR}(h)$ は Quick Filter 部で絞り込まれる partition の割合で、式 (2) で表される [3]。

$$\mathrm{PAR}(h) \approx \left(1 - \frac{m_q}{2f}\right)^h \quad (2)$$

BSQF と通常の BSSF の検索コストを、図 2に示す。

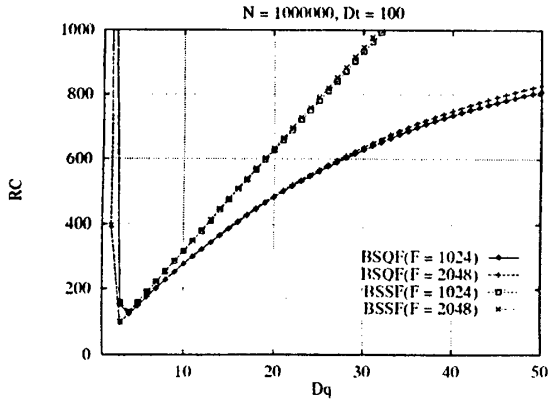

図 2: $N = 1,000,000$　$D_t = 100$ の検索コスト

4.2 スマート検索

スマート検索 (*smart retrieval*) とは、[1] で提案された BSSF における問い合わせ最適化の手法である。スマート検索を BSQF にも応用した場合の検索コストを、図 3に示す。

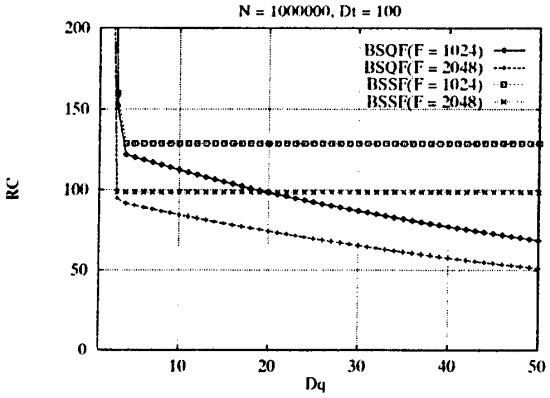

図 3: スマート検索を行なった場合の検索コスト

4.3 格納コスト

図 2で $F = 1024$ の場合、通常の BSSF の格納コストは 17361 ページ、BSQF の格納コストは 17423 ページとなり、そのオーバーヘッドは 1% 以下と非常に小さい。

5 まとめ

本稿では、シグネチャファイルの新しい構成法である BSQF の提案を行ない、その検索コストについて見積り式を示し、評価を行なった。また、格納コストについても考察を行なった。

その結果、BSQF の検索コストは通常の BSSF より悪くなることはなく、一般に D_q が大きいほど BSQF が有利になることが分かった。また、格納オーバーヘッドは非常に小さいことが分かった。さらにまた、スマート検索を行うことにより、さらにコストを低減させることができることを示した。

今後の課題としては、$T \supseteq Q$ 以外の問い合わせ条件についての評価などがあげられる。

参考文献

[1] Yoshiharu Ishikawa, Hiroyuki Kitagawa, and Nobuo Ohbo. Evaluation of Signature Files as Set Access Facilities in OODBs. In *Proc. ACM SIGMOD*, pp. 26–28, 1993.

[2] Zezula P., Rabitti F., and Tiberio P. Dynamic Partitioning of Signature Files. *ACM Trans.*, Vol. 9, No. 4, pp. 336–369, 1991.

[3] Ciaccia P. and Zezula P. Estimating Accesses in Partitioned Signature File Organizations. *ACM Trans.*, Vol. 11, No. 2, pp. 133–142, 1993.