1P－2

シグネチャキャッシュ －分散環境における情報資源アクセスの効率化手法－

石川 佳治† 北川 博之‡

†奈良先端科学技術大学院大学 情報科学研究科 ‡筑波大学 電子・情報工学系

ishikawa@is.aist-nara.ac.jp kitagawa@is.tsukuba.ac.jp

1 はじめに

近年のインターネットの発展にともない，ネットワーク上に莫大な情報資源が存在することとなった．このようなネットワーク上の情報資源を有効に活用するためには，情報資源への高速なアクセス手法が提供されねばならない．本稿では，分散情報資源アクセスの効率化のための手法としてシグネチャキャッシュ (SignatureCache) を提案する．この手法では，まず，それぞれの情報資源の内容がコンパクトなビット列であるシグネチャ (signature) で表現される．ネットワーク上の各クライアントではこれらのシグネチャがキャッシュされ，一種の索引としてはたらき，効率的な検索処理が可能となる．本稿ではシグネチャキャッシュのアイデアと，問合せの包含関係を用いた効率的な問合せ処理について述べる．

2 シグネチャファイルについて

まず，シグネチャキャッシュの基となったシグネチャファイルの手法について説明する．**シグネチャファイル** (signature file) とは，おもにテキスト検索の分野で用いられてきた索引手法である[Fal92]．シグネチャ (signature) は，個々の文献の内容をコンパクトに表現した固定長 (Fビットとする) のビット列であり，スーパーインポーズドコーディング (superimposed coding) と呼ばれるコーディング手法が用いられることが一般的である．スーパーインポーズドコーディングでは，文献中に現れるそれぞれの単語について，まずハッシュ関数により Fビットの**単語シグネチャ** (word signature) が作成される．単語シグネチャでは Fビットのうちパラメータ m で決められた数だけ“1”が立てられる．次いで，文献中の各単語に対する単語シグネチャのビットごとの論理和をとることにより**文献シグネチャ** (document signature) が得られる．このようすを図1に示す．それぞれの文献について，文献シグネチャとその文献番号の組を格納したものがシグネチャファイルである．

単語	単語シグネチャ
distributed	00010001
information	10010000
文献シグネチャ →	10010001

図 1: シグネチャの生成 ($F = 8, m = 2$)

シグネチャファイルを用いた検索では，まず，問合せとして与えられた単語の集合から文献シグネチャ作成と同様の手法で**問合せシグネチャ** (query signature) が作成される．次いで，問合せシグネチャとシグネチャファイル中の各文献シグネチャとのパターンマッチが行なわれる．文献シグネチャは，問合せシグネチャにおいて“1”の値をもつビット位置すべてについて“1”の値をとるとき問合せ条件を満たす候補となり，**ドロップ** (drop) と呼ばれる．しかし，ハッシュ関数の衝突やスーパーインポーズドコーディングの影響により文献シグネチャが誤ってパターンマッチしてしまうこともある．ドロップのうち実際には問合せ条件を満たさないものを**フォルスドロップ** (false drop) と呼び，満たすものを**アクチュアルドロップ** (actual drop) と呼ぶ．

SignatureCache: An Efficient Access Method for Distributed Information Resources

Yoshiharu Ishikawa† and Hiroyuki Kitagawa‡

†Nara Institute of Science and Technology

‡University of Tsukuba

3 シグネチャキャッシュ

3.1 前提とする環境

本研究では，ネットワーク上に多数の情報資源サーバが分散して存在する環境のもとでの検索処理の効率化を考えているが，ただし，そこでは**仲介機構** (mediator) の存在を仮定している．仲介機構の役割の一つは，個々の情報資源サーバが提供する情報資源からそれぞれの情報資源の内容を抽出し，シグネチャを作成することである．具体的な例として World-Wide Web (WWW) を考えると，仲介機構はそれぞれの情報資源サーバ，すなわちWWWサーバが提供するページから単語情報を抽出してシグネチャを構成し，そのページの URL (Unique Resource Locator) と組にして仲介機構が管理するシグネチャファイルに格納する．ただし，仲介機構自体には情報資源 (WWWページ) そのものは格納されない．以下ではWWWを想定して説明するが，情報資源はWWWに限らない．

3.2 シグネチャキャッシュの概念

シグネチャキャッシュ (SignatureCache) は，情報資源を検索する各々のクライアント上で管理される一種のシグネチャファイルである．しかし，それが実際には仲介機構で管理されるシグネチャファイルのキャッシュであることや，それぞれのシグネチャがどのような問合せによってキャッシュされたか，また，各シグネチャがアクチュアルドロップであったかフォルスドロップであったかなどの付加情報を管理する点で異なっている．シグネチャキャッシュの管理方式としては，シグネチャのそれぞれのビット位置ごとに別々にファイルを作成する**ビットスライストシグネチャファイル** (Bit-Sliced Signature File, BSSF) 方式[Fal92]を想定する．

例をもとにシグネチャキャッシュの概念を説明する．クライアント上で以下の問合せQ1が発行されたとする．ただし，現在クライアント上にはシグネチャは一切キャッシュされていないことを仮定する．

```
Q1: FIND database distributed information
```

この問合せは，単語の集合 {database, distributed, information} のすべてに関連する情報資源を検索するものである．処理のステップを以下に示す．

1. クライアントは，まずこれら三つの単語から問合せシグネチャ (10010011とする) を作成する．これが仲介機構に送られる．
2. 仲介機構ではシグネチャファイルに対しパターンマッチを行ない，マッチしたシグネチャとその URL の組をクライアントに返す．クライアントはこれをキャッシュする．
3. ステップ2において得られた URL をもとに，対応する情報資源を検索する．
4. 得られた情報資源が実際に問合せ条件を満たすかどうかをチェックする．つまり，フォルスドロップレゾリューションの処理を行なう．

5. 問合せ条件を満たした情報資源をユーザに返す.
6. キャッシュされたシグネチャに対し, 問合せQ1 に対し実際にはアクチュアルドロップであったかフォルスドロップであったかを示す情報を付加する.

問合せ結果のシグネチャキャッシュのようすを図 2 に示す. ステップ 2 の結果により 6 つのシグネチャが得られているが, ステップ 4 によりURL#2, URL#6 の情報資源は実際には問合せ条件を満たさないことがわかったため, ステップ 6 においてフォルスドロップであるとしてf が記入されている. 残りのシグネチャについてはアクチュアルドロップであったためa が記入されている. この情報は今回の問合せ処理の効率化に関しては効果はないが, その後同様の問合せを処理する場合に効力を発揮する. たとえば, Q1 とまったく同じ問合せが与えられれば, a とマークされたアクチュアルドロップだけ検索すればすみ, シグネチャどうしのパターンマッチやフォルスドロップレゾリューションは必要ない.

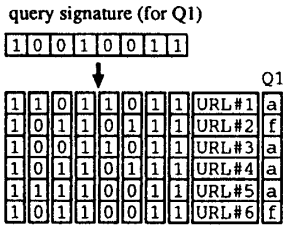

図 2: シグネチャキャッシュ

4 問合せの関連を活用した問合せ処理

シグネチャキャッシュの特徴は, 単に同じ問合せが与えられた場合でなくとも, ある種の関連をもつ問合せが与えられた場合にも活用可能であることである. その例を以下に示す.

4.1 活用例 1

図 2 のシグネチャキャッシュが存在するとき, 次の問合せQ2 を考える.

```
Q2: FIND database distributed information retrieval
```

この問合せはQ1 のキーワード集合に'retrieval' を付け加えたものであり, 検索をより詳細化したものである. よって, Q2 に対しドロップとなるシグネチャは, すべてすでにキャッシュされていることになる.

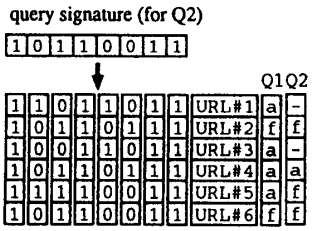

図 3: 包含関係の活用例 (1)

問合せの結果のシグネチャキャッシュを図 3 に示す. 'retrieval' に対する単語シグネチャが'00100001' であったとすると, Q1 の問合せシグネチャとの論理和をとった'10110011' がQ2 の問合せシグネチャとなる. URL#1, URL#3 に対するシグネチャは, 問合せシグネチャに対するパターンマッチ条件を満たさないためドロップとならない (-で示す). 残りのURL#2, URL#4, URL#5, URL#6 がパターンマッチの条件を満たすが, URL#2, URL#6 についてはQ1 のフォルスドロップレゾリューションの結果, {database, distributed, information} を含んでいないことが明らかとなっている. よって, 不要なフォルスドロップレゾリューションを避けることができ, URL#4, URL#5 についてのみフォルスドロップレゾリューションを行なえばよい.

4.2 活用例 2

今度は関係が逆の場合を考える. 図 2 のQ1 に対するシグネチャキャッシュがあったとき, 次の問合せQ3 を考える.

```
Q3: FIND distributed information
```

この場合も, Q1 に対するシグネチャキャッシュを利用することができる. Q1 に対するシグネチャでアクチュアルドロップであるとマークされているもの (URL#1, URL#3, URL#4, URL#5) については {database, distributed, information} をすべて含むわけであるから, 当然Q3 についてもアクチュアルドロップとなる . よって, これらについてはフォルスドロップレゾリューションは不要である. 一方, Q1 に対するシグネチャでフォルスドロップとマークされたもの (URL#2, URL#6) については問合せを満たす可能性があるためフォルスドロップレゾリューションが必要である. 以上の処理を通じて, シグネチャ部分にアクセスする必要はなく, URL ファイルとQ1 に対するフォルスドロップレゾリューションの結果のみにアクセスすればよいことに注意が必要である.

しかし, この場合には, Q1 に対するキャッシュに含まれないシグネチャを仲介機構から検索する必要がある. すなわち, Q3 の問合せシグネチャにマッチするシグネチャのうちで, Q1 の問合せシグネチャににマッチしないものを新たに検索しなければならない. しかし, この検索ではQ1 との差分だけを検索すればよいため, 検索処理は効率的である. Q3 の問合せ処理後のシグネチャキャッシュの内容を図 4 に示す. URL#7, URL#8 が新たに検索したシグネチャである.

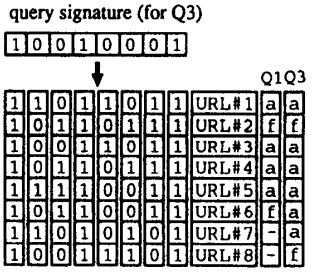

図 4: 包含関係の活用例 (2)

5 まとめと今後の課題

本稿ではシグネチャキャッシュの概念とその特徴について述べた. シグネチャキャッシュは分散したインクリメンタルな索引[RES93]の一種とも考えられるが, シグネチャファイルが単純な構造を持つことより木構造の索引と比べ柔軟性に富んでいる. 今後はシグネチャキャッシュの管理アルゴリズムと問合せ処理の詳細化, およびその性能評価を行ないたい.

参 考 文 献

[Fal92] C. Faloutsos: "Signature Files", in W. B. Frakes and R. Baeza-Yates eds., *Information Retrieval - Data Structures and Algorithms*, chapter 4, pp. 44–65, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[RES93] N. Roussopoulos, N. Economou, and A. Stamenas: "ADMS: A Testbed for Incremental Access Methods", *IEEE Trans. on Knowl. and Data Eng.*, 5(5):762–774, Oct. 1993.