索引の動的ロードによる全文検索方式の高速化

4N－2

和田久美子 池田恵美 森田幸伯
{kwada,ikeda,morita}@okilab.oki.co.jp
沖電気工業(株) メディアネットワーク研究所

1　はじめに

電子メディアの著しい普及に伴い，様々な分野で大規模な文書の電子化が急速に進んでいる．次世代電子図書館システムの構築において，大規模電子化文書に対する効率のよいテキスト検索技術は不可欠である．

全文検索技術は，テキストの形態やデータベース構造等に依存せず実用的な検索を実現できる点で非常に有効であると考えられる．しかし，検索対象は日々急速な勢いで増加しており，非常に大規模な検索対象に対しても実用的な速度で検索が可能であることが要求されている．

本発表で述べる全文検索システムでは，可変長の疑似語句を索引語として登録する方式を採用している．検索を高速化するため，語句の先頭部分に対応するインデックス(以降先頭インデックスと呼ぶ)をメモリ上に展開する．しかし，大規模な本文データに対する索引では，語句集合も大規模なものになるため，メモリ上に展開される先頭インデックスのメモリ量も大規模になりがちである．このような状況では，検索演算処理にはさらにメモリ上に広い作業領域が必要とされるため，先頭インデックスはできるだけ小さく実現することが望まれる．

語句集合は，本文データの性質によって偏りがあることから，本発表では，大規模な検索対象に対しても有効に検索が可能となるよう，語句の先頭の文字並びに関する頻度情報や文字種をもとに，メモリ上へのロード部分を動的に決定する手法について述べる．

2　先頭インデックスのメモリ展開の問題点

インバーテッドファイル形式で格納されている索引語ファイルを高速に検索するには，文字単位の多階層インデックスを構成するのが効果的である．

索引語の先頭の固定文字数に対して一定階層数のインデックスを用いれば実現は簡単だが，検索速度にばらつきが生じる．たとえば，カタカナ文字や英字から成る専門用語などは比較的文字列長が長いため，さらに絞り込みを行うためのファイルアクセスが必要となり，充分な検索速度を得ることができない．階層数を増加させればこの問題は解決できるが，使用可能なメモリ量には限度があるため，本文データ量が大規模の場合には解決にならない．

例えば，索引語の先頭の n 文字を用いて，階層数が一定の値 n をとる先頭インデックスを構成する場合，本文データサイズと先頭インデックスのメモリ展開サイズの関係は図1に示すような関係にある．環境にも依存するが，大規模データに対する検索では演算処理などに要する作業領域を大きく取る必要があるため，$n=2$ を越える先頭インデックスをメモリ上に展開するのは困難であると思われる．

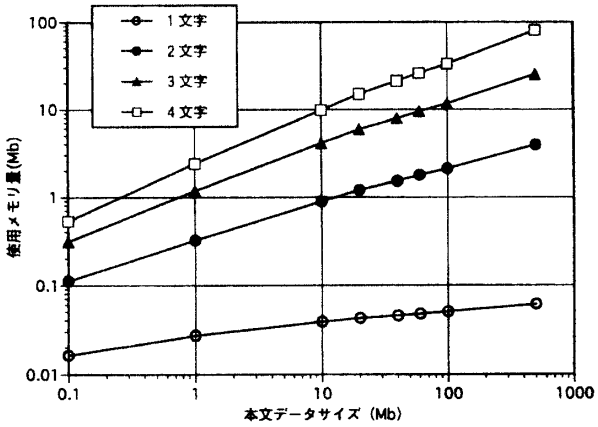

図 1: 先頭インデックスの展開文字数とメモリ量の関係

3　先頭インデックスのメモリ展開方式

先頭インデックスのメモリ展開方式について説明する．なお，システム概要については [1] 参照．

3.1　索引生成時処理

まず，全文検索用索引生成時に，先頭インデックスとしてメモリ展開する可能性のある文字数の最大値 n を指定し，索引語の先頭 n 文字に関する情報(以降先頭インデックス情報と呼ぶ)をあらかじめ索引ファイルに格納しておく．先頭インデックス情報は，先頭 n 文字が指定された文字並びで開始されるもっとも小さい(辞書式順序による)索引語に対する出現位置情報の格納オフセットである．n の値を大きく取るほど目標索引語に対応する出現位置情報を高速に検索できる．

Implementing fast full text search system by dynamically expanding indexes on memory
Kumiko Wada, Emi Ikeda and Yukihiro Morita
Oki Electric Industry Co., Ltd., Media Network Laboratories

メモリサイズ制限によりメモリ上に展開できない場合でもなるべく性能を保つため，先頭インデックス情報のファイルへの格納形式は B 木構造である．これにより，索引情報の更新処理にも対応することができるようになっている．先頭 n 文字に対するインデックスのうち，メモリ上に展開されない部分はファイルアクセスによって検索される．

3.2 先頭インデックスのメモリ展開処理

次に，全文検索用サーバプロセスの起動時に，先頭インデックスを指定されたサイズでメモリ上に展開する．索引語の最大 n 文字までの先頭文字並びに関する先頭インデックスを展開することができる．メモリ展開時には，i 文字めまでが同一文字並びから成る索引語集合の $i+1$ 文字め以降の異なり数をプライオリティキューで管理し，一定メモリ量を越えた場合は，もっとも異なり数の少ない部分に関する部分インデックスをメモリ上から必要メモリ量に対応する分だけ削除する．

先頭インデックスの各階層は，文字コードと次の階層へのポインタの対応を格納したハッシュ表である．次の階層がメモリ上にない場合は，ファイルへのオフセット値が格納されている (図 2)．

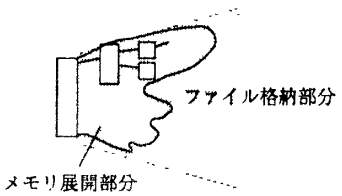

図 2: 先頭インデックス概念図

3.3 検索時処理

検索時には，まずメモリ展開された先頭インデックスから検索を開始し，各階層を手繰りながら必要ならばファイル上の先頭インデックスを検索して最終的に目標索引語の出現位置情報の格納場所を獲得する．

4 評価

以上で述べた方式にしたがって先頭インデックスをメモリ上に展開した場合の索引ファイルのアクセス回数を，固定階層数インデックスでファイル上の先頭インデックスを用いない場合と比較したものを図3に示す．ここでは，検索文字列が与えられてからそれに対する出現位置情報を獲得するまでにファイルアクセスしたブロック数を測定した．評価に用いた本文データは，特許公開公報約 23,800 件 (500Mb)，動作環境は HP9000/777(メモリ 128Mb) である．

図4は，指定されたメモリ量で先頭インデックスを展開した場合のファイル上の先頭インデックスの平均アクセス回数を比較したものである．同サイズのメモリを使用した場合，固定階層数の先頭インデックスを展開した場合に比べて動的展開方式の方が，アクセス回数が減少していることがわかる．

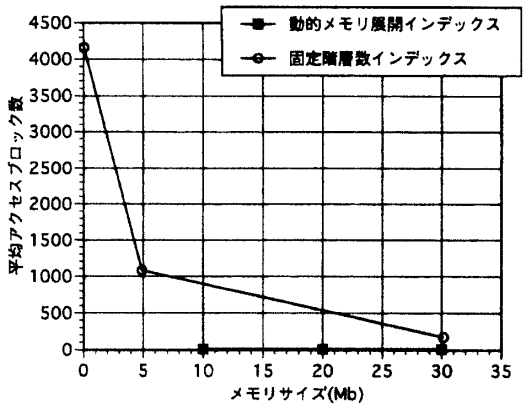

図 3: 平均アクセス数

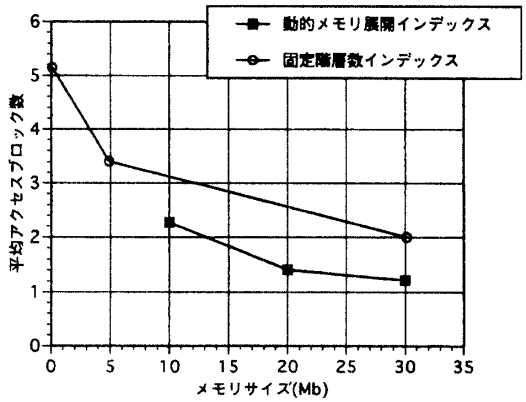

図 4: ファイル上の先頭インデックスへの平均アクセス数

5 おわりに

疑似語句抽出方式による全文検索システムにおいて，先頭インデックスの動的メモリ展開方式を示した．今後は Gb オーダの索引に対する検証を行う予定である．また，語句の文字種やその他の性質を用いたチューニングを行う必要がある．本研究は，日本情報処理開発協会殿の次世代電子図書館システム研究開発事業の一環として行われている．

参考文献

[1] 池田他: “疑似語句抽出による大規模日本語全文検索方式”, 第 55 回情処全大, 4N-02, 1997.

[2] 森田他: “疑似語句抽出による大規模日本語全文検索方式”, 第 10 回ディジタル図書館ワークショップ,URL http://www.DL.ulis.ac.jp/DLworkshop/