情 報 学 基 礎 25－1
（１９９２．５．１２）

全文検索の技術動向とシステム事例

菊地 芳秀、 小川 隆一、 高橋 恒介、 杉本 欽一、 金田 悟

日本電気株式会社

全文検索を高速化するための最近の技術動向について、ソフトウェア面およびハードウェア面から概観する。ソフトウェア面では、照合アルゴリズムそのものの研究よりも実用化技術の研究が盛んになってきており、検索の加速化を中心とした最近の動向について述べる。ハードウェア面では、速度面で有利なセルラーアレイ方式と検索文字列の容量面で有利な有限状態オートマトン方式を中心に述べる。最後に、検索ハードウェアとディスクアレイから構成される検索システムを試作した結果について報告する。

Full Text Retrieval Technologies' trend and System Implementaion

Yoshihide KIKUCHI, Ryuichi OGAWA, Kousuke TAKAHASHI, Kinichi SUGIMOTO, Satoru KANEDA

NEC Corporation

1-1, Miyazaki 4-Chome, Miyamae-ku, Kawasaki, Kanagawa 216, Japan

Recent developments in fulltext retrieval technologies are surveyd both from software and hardware viewpoins. As for software technologies research emphasis has shifted from pattern matching algorithm to implementation issues such as accelaration methods for retrieval, which are described in detail. As for hardware technologies two prevailing ones, cellular array and finite state automaton, are described. Also an implementation example of a high-effecient fulltext retrieval system, composed of retrieval hardware and disk array, is reported.

１．はじめに

パーソナルコンピュータやワードプロセッサなど情報機器の普及、および出版工程の電子化などに伴ない、大量の文書が電子化され、フルテキストデータベースとして蓄積されてきている。フルテキストデータベースとは、文書検索において、いわゆる二次情報（題名、著者名、発行日などの書誌情報やキーワード、抄録）だけでなく、一次情報（本文全体）を記録し、これを参照できるデータベースのことをいう。このため、一次情報の中から高速に目的のキーワードを探し出す全文検索技術の確立が求められている。

２．ソフトウェアによる全文検索技術

ソフトウェアによる全文検索技術は、大きく２つに分けることができる。１つは照合アルゴリズムそのものの研究であり、もう１つはインデックスファイルなど検索用の補助ファイルを作成することにより全文検索の高速化を図る研究である。

照合アルゴリズムの研究は古くからなされており、1970年代後半には現在広く利用されている照合アルゴリズムがほぼ確立している[1]。その代表的なものとしては、次の３つの方式が挙げられる。

①Knuth-Morris-Prattアルゴリズム[2]
②Boyer-Mooreアルゴリズム[3]
③Aho-Corasickアルゴリズム[4]

照合アルゴリズムの研究は、これらのアルゴリズムを基に改良が続けられているが、検索時間そのものを縮めることは難しくなってきており、研究の方向は正規表現の照合やあいまい照合へと移りつつある[5]。

一方、検索補助ファイルを作成する方式については、事前に準備しておく時間が必要、本文と別にかなりな量のファイル容量が必要になるなどの課題はあるが、検索自体は非常に高速になることから、実用化へのアプローチとして研究が進められている。

その代表的な方式を次に挙げる。

①文字成分表方式
②全文検索用インデックス方式
③圧縮ファイル方式
④学習方式

この中で、①③は、照合アルゴリズムまたは照合ハードウェアの併用を前提とし、検索の加速化に用いられている。また、②④は逐次照合をしないで全文検索を行う方式である。以下、これらの方式について簡単に紹介する。

2.1 文字成分表方式[8]

本文を逐次検索する前に検索キーワードが文書内に無いことが分かれば、その文書の検索をしなくて済み効率的である。このような考えから作られたのが文字成分表である。

文字成分表では、文書中に現れる文字を１、現れない文字を０として文字の有無を文書毎に表にしている（図１）。利用者は文書を登録する際、事前に文字成分表も作成しておく。検索キーワードを構成する文字を文字成分表からサーチすることにより、検索キーワードが含まれない文書を排除することができ、検索対象となる文書を絞り込むことができる。

	あ	い	う	・	・	・	検	・	・	索	・	・	・
文書1	1	1	0	・	・	・	1	・	・	1	・	・	・
文書2	0	1	0	・	・	・	1	・	・	1	・	・	・
文書3	0	0	0	・	・	・	1	・	・	1	・	・	・
・・	・	・	・	・	・	・	・	・	・	・	・	・	・
・・	・	・	・	・	・	・	・	・	・	・	・	・	・
文書N													

図１ 文字成分表

この方式では、

①文書が大きいと文字成分数が多くなり、絞り込みがしにくくなる。

②文書が小さいと文字成分表の大きさが文書に較べて無視できなくなる。

などの課題がある。このうち、②に対しては、文字成分表を折り畳んで使用する方法がある（図２）。こうすることで精度は落ちるが、記憶領域の縮小化、サーチ速度の向上に効果がある。

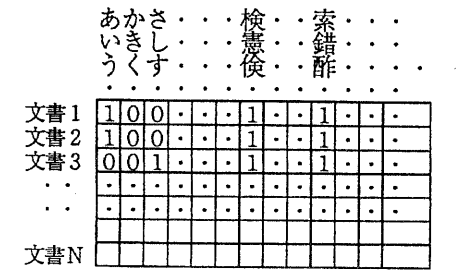

図２ 圧縮文字成分表

2.2 全文検索用インデックス方式

通常の検索用インデックスは、予め定められたシソーラスに対して作成するが、これだとシソーラスにないキーワードは検索できず、全文検索には向かない。全文検索用のインデックスとしては、テキスト中に現れる全文字の位置情報を持ったインデックスが必要となる（図３）。

	出現位置情報
・	・・・・・・・・・・・・・・・
拡	461、774、1091、・
・	・・・・・・・・・・・・・・・
張	462、826、1092、・
・	・・・・・・・・・・・・・・・

図３ 全文検索用インデックスの例

検索方法としては、検索したいキーワードを文字種に分解し、全文検索用インデックスから文字種の出現位置情報を得る。位置情報を基に各文字のつながりをチェックし、キーワードの通りに並んだ箇所があれば、そこがキーワードの存在する部分となる。

文字間の接続をチェックする手間を削減する方法としては、予め２文字以上のインデックスを作成する方法が考えられるが、インデックスの容量が膨大になるという問題も生じる。

この容量と接続計算時間のトレードオフ問題に関しては、日本語テキストの特徴を考慮して、仮名文字を含むもののみ２文字にした実験が報告されている[6]。

全文検索用インデックス方式では、インデックスの量が膨大になるという欠点はあるが、処理が簡単で、かつ、キーワードの自動抽出で問題になるような「キーワード漏れ」を防ぐことができる。応用としては、テキストの読み出し速度が遅く、本文より大きくなるインデックスを納める余裕のあるＣＤ－ＲＯＭの全文検索に向いた方式であると言える。

2.3 圧縮ファイル方式

全文検索とキーワード検索の中間に位置する検索手段として、全文テキストを一定の方式で圧縮したシグネチャーファイルを作成し、これを対象としたパターン照合を行なう方式がある[7]。シグネチャーファイルもインデックスと同様に、記憶容量の点やデータ更新の手間から見ると必ずしも効率的な検索方式ではないが、全文検索の負荷を軽減するのに効果的である。

シグネチャーファイルは、作成方法によって２つに分けられる。１つは、各文字のビットを削除するなどして曖昧度を増加させることによりデータの圧縮を図るものである[9]。もう１つはテキストサーチマシンＴＳＭ－Ｉに用いられている凝縮本文[8]のように、助詞

・接続詞等の付属語や繰り返し現れる単語など、余分な情報を削除することでデータの圧縮を図るものである。

圧縮率を高くすると、前者の場合ではゴミを拾う確立が高くなり、後者の場合では検索漏れの確立が高くなる。

2.4 学習方式

ニューラルネットワークまたはそれに類した学習ファイルを事前に作成することにより検索の高速化を図る方式が2社から発表されている。どちらも不完全一致検索のしやすさが特徴となっているが、詳細は不明である。

3．ハードウェアによる全文検索技術

フルテキストサーチにおけるハードウェアアーキテクチャの研究は、記号処理技術の研究に関連して古くからなされている[10]。この頃に知られていた方式として、並列照合方式、セルラーアレイ方式、有限状態オートマトン方式などがある[11]。最近では、ＬＳＩ製造技術の進歩に伴い、これらを組み合わせたプログラマブル順序論理方式やダイナミックプログラミング方式が新たに提案され、ＶＬＳＩ化されている[12]。

ここでは、検索ハードウェアとしてもっとも単純な並列照合方式について述べ、検索速度の点で優れているセルラーアレイ方式と、正規表現等、検索の柔軟性の高い有限状態オートマトン方式について概説する。

3.1 並列照合方式

比較器を多数並べて、テキストと検索文字列の照合を並列に行う方式であり、ソフトウェアでのもっとも素朴なアルゴリズム(Naive法)に対応する。図４に動作原理を示す。各セルの一致信号の論理積を取ることによって全体の一致結果を発生する。一般的には回路規模を小さくするために、レジスタと比較器の代わりにＣＡＭ(Content Addressable Memory)が用いられる。

並列照合方式では考え方が単純な代わりに、使用ハード量が多い、ＶＬＤＣ(Variable Length Don't Care)文字が扱えないなどの課題があるため、このままでは使われないが、他の方式の考え方のベースとなっている。

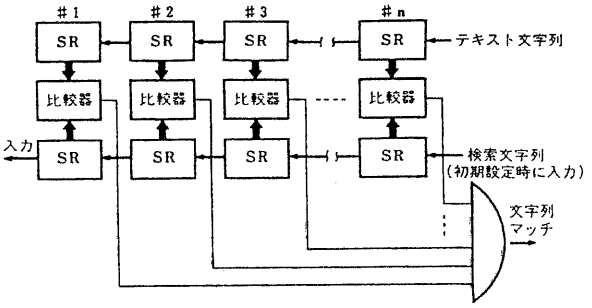

図４ 並列照合方式の動作原理

3.2 セルラーアレイ（ＣＡ）方式

１文字単位の比較器とロジックが対になったセルをアレイ状に配した照合方式をセルラーアレイ(Cellular Array)方式と呼ぶ。各セルから出力される一致信号と、それまでの一致結果との論理積を順番に取ることで、まとまった文字列の照合を行う。この方式はテキストの与え方の違いにより、ブロードキャスト型[13]とシストリックアレイ型[14]とに分かれる。現在、シストリックアレイ型はほとんど使われていないため、ここではブロードキャスト型について述べる。

ブロードキャスト(broadcast)型の動作原理を図５に示す。入力されるテキストは１文字毎に全てのセルへ一斉に入力され、比較が行われる。比較器から出力された一致信号と前のセルの一致信号との論理積が取られ、次のセルに伝搬される。最後のセルから出る一致信号が全体の一致結果を表す。

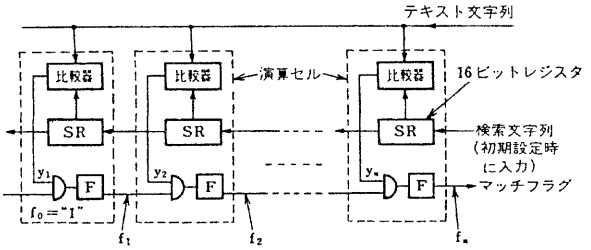

図5 ブロードキャスト型の動作原理

比較結果が伝搬される部分は順序論理回路でもあるので、この方式はプログラマブル順序論理回路方式と考えることもできる。照合部分にＣＡＭを用い、順序論理回路に曖昧照合機能を持たせたものが報告されている[15]。

セルラーアレイ方式では、検索キーワードの容量面で制限はあるが、照合が一斉に行われることから登録しているキーワードの数が増えても照合速度が一定であるという利点がある。

3.3 有限状態オートマトン（ＦＳＡ）方式

有限状態オートマトン(Finite State Automaton)方式は、テキストの入力によって状態を遷移させながら照合を行う方式である。

状態遷移をエミュレートする機構としては、状態番号と入力文字コードで表された２次元の状態遷移表をＲＡＭに格納して次の状態を得る方法[16]と、テキストとキーワードとを比較器で比較してその成功／失敗により次の状態を求める方法とがある。前者の場合、照合速度は速いが、２次元の状態遷移表をＲＡＭに格納すると膨大なメモリ量が必要になるという欠点がある。また、後者の場合、前者に較べて使用メモリは少ないが、照合時間がかかるという欠点がある。これに対し、照合条件の負荷に応じて両者を使い分ける例が報告されている[17]。また、前者の場合は、連想メモリ（ＣＡＭ）を使うことによりハードウェアの量を抑えることもできる。

ＦＳＡでＶＬＤＣ文字を扱う場合、そのままだと非決定性ＦＳＡになるため状態遷移先が複数個となる。Haskinはこれに対し、状態遷移先が１つになるように状態遷移表を分割する方法[18]を提案している。

ＦＳＡ方式を用いると、遷移関数を格納する領域が大きい、前処理時間がかかる、などの欠点はあるが、正規表現を用いた複雑なパターンの検索が可能となるなど魅力も大きい。

セルラーアレイ方式との比較では、検索キーワード数が増えると照合時間が延びるという問題はあるが、状態遷移表をプロセッサの外部に持たせることができるため、検索キーワード数を多くとることができるという利点がある。

4．システム事例

このような状況の中、筆者らは検索ＬＳＩを利用した検索システムをいくつか試作してきた[19][20]。現在、ディスクアレイを内蔵したサーチマシンを試作し、基本的な実験を進めている。以下、今回試作した検索システムの概要および実験結果について報告する。

4.1 システムの構成と概要

検索システムは、ＥＷＳおよびそのバックエンドプロセッサであるサーチマシンからなる。さらに、サーチマシンは８台のハードディスクからなるディスクアレイと、検索ＬＳＩを用いた２枚の検索ボードから構成されている。

サーチマシンは、ディスクアレイと検索ボードとを内部バスにより直接結合するアーキテクチャを採用している。すなわち、テキストはＥＷＳを介さずに検索が行われる。ホストのＥＷＳとのやりとりは、検索キーワード

の受け渡し、検索テキストの検索範囲指定、検索結果の受け取りなど最小限のものになっている。

図６に検索システムの構成を、図７にシステムの外観を示す。

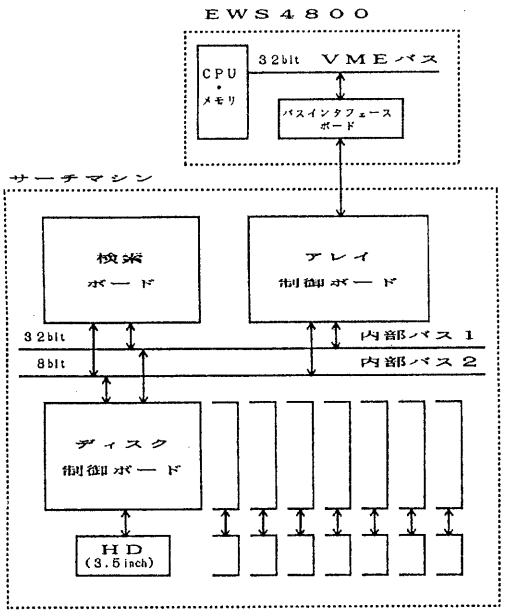

図６ 検索システムの構成

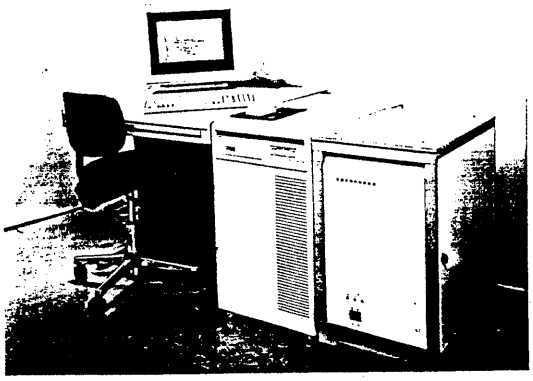

図７ 検索システムの外観

検索は、次の手順で行われる。

①検索ハードウェアに検索キーワードを登録する。

②被検索ファイルのディスク内の位置を解析し、検索したい部分の読み出しをディスクアレイに指示し、検索を開始する。

③検索結果を検索ハードウェアから引き取る。

今回、ディスクアレイの性能を引き出すため、テキストファイルは細かく分割せず、各テキストをつなぎ合わせた大きなファイルを作成している。このため、条件検索に対しては、各テキストの範囲を規定する記号またはキーワードも一緒に検索する方式を採っている。条件判断は、検索結果を引き取ったのちにソフトウェアで処理している。

4.2 検索ボード

検索部は、ＶＭＥダブルハイトボード２枚からなり、心臓部には検索ＬＳＩ：ＩＳＳＰ(Intelligent String Search Processor)[15]を２つ用いている。ＩＳＳＰの主な特徴は次の通りである。

- ６４個のキーワードの並列検索。
- １０M文字／秒のテキスト入力速度。
- 曖昧文字列検索（１文字の誤り、欠け、混入を許容）。
- 可変長ワイルドカード機能。
- キーワード数の拡張が容易。

検索ボードでは、ＩＳＳＰの持つ上記機能に加え、以下のような特徴を持たせている。

- ２KWの結果メモリを持ち、一致結果を２０００件まで生成／格納する。
- 検索ＬＳＩの前段にユーザが内容を指定できる４つの読み飛ばしフィルタを設け、スペースやタブ、リターンなどの記号を除外した検索を可能とした。これは、整形のためにスペース・タブ・リターンなどがキーワード中に挿入された文章でも、検索が行えるようにするためである。

検索ボードの仕様は、現在次のようになっている。

・検索キーワード数：最大１２８個
・入力データ幅　　：１６bit
・テキスト入力速度：最大８Mbyte／秒
・一致結果格納容量：２０００件

図８に検索ボードの構成を示す。

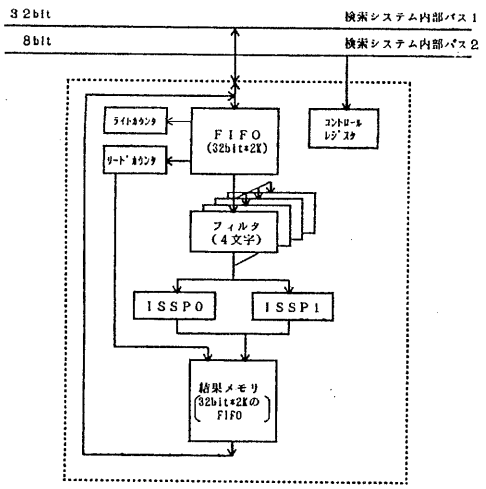

図８ 検索ボードの構成

4.3 ディスクアレイ

ディスクアレイは８台の３．５インチ型ハードディスクから構成されており、８００Mbyteの容量を持つ。ＥＷＳとはＶＭＥバスで接続され、デバイスドライバにより通常は他のディスクと同じように認識される。

本ディスクアレイはディスク制御ボード上にバッファを持ち、各ディスクから読み出されるデータを一旦バッファに格納している。これにより、８台のディスクは非同期で動作させることができ、ディスクユニットは市販の安価なものが使用可能となっている。

また、バッファにデータを格納していることから、検索ボードへデータを転送する際、３２バイト単位で転送範囲を設定できる。これにより、大容量テキストファイルのある部分からある部分までというような部分検索が可能となっている。

4.4 実験と考察

今回、本検索システム上にUNIXのコマンドである"grep"と似た機能を持つコマンドを作成して評価実験を行った。使用したテキストファイルの長さは、約１０Mbyteであり、一致箇所はこの中で数カ所となるようなキーワードを選んだ。

検索コマンドにおいて、一致件数のみの表示としたとき、検索スピードは、平均で２Mbyte／秒となった。これは１秒間にＡ４文書（１０００文字）１０００頁の検索速度である。

厳密な測定は行っていないが、極端に一致が多い場合を除き、検索速度の大部分がディスクの読み出し速度で決まっていると考えられる。検索ハードウェアの性能を引き出すためには、ディスクアレイの平均読み出し速度をさらにアップする必要がある。

ただ、ニーズ調査において数百Mbyteの検索要求も生じてきているため、検索の加速化を含めシステム自体の高速化を再検討する必要があると考えている。

５．むすび

フルテキストデータベースの技術動向について、ソフトウエア／ハードウエアの両面から概観し、検索ハードウェアとディスクアレイからなる検索システムの試作結果について報告した。

現在、検索ハードウェアの性能を十分に引き出すような周辺環境が高価なため、ソフトウェアによる検索の研究が盛んになってきていると言えるが、これはとりもなおさず検索に対するニーズの高さを物語るものでもある。

今後、検索システムの実用化のための検討

を行っていくと共に、検索システムを利用する側に立ったアプリケーションの検討も進めて行きたい。

６．参考文献

[1] 竹田：固定文字列と文字種の混在するパターンのを対象としたAho-Corasick型パターン照合機械の構成法，九州大学大型計算機センター，計算機科学研究報告第６号，pp.29-51 (1989)

[2] Knuth, Don, H.J. Morris, and V.R. Pratt: Fast Pattern Matching in Strings, Technical Report STAN-CS-74-440, Sci. Dep.,Stanford University (August 1974)

[3] Boyer, Robert and J.S. Moore: A Fast String Searching Algorithm, Communications of ACM, Vol.20, No.10, pp.762-772 (October 1977)

[4] Aho, Alfred, and Margaret Corasick: Efficient String Matching: An Aid to Bibliographic Search, Communications of the ACM, Vol18, No.6, pp.333-340 (June 1975)

[5] 竹田：全文テキスト処理のための高速パターン照合アルゴリズム，1991年情報学シンポジウム予稿集，pp.85-96(1991)

[6] 菊池：文字列照合を用いた全文検索における仮名文字検索の高速化手法，情処研報，Vol.91, No.46, 91-DBS-83 (May 1991)

[7] Faloutsos, Christos: Access Method for Text, Computing Surveys, Vol.17, No.1,pp.49-74 (March 1985)

[8] 加藤，藤澤，大山，川口，畠山，兼岡，秋沢：大規模文書データベース用テキストサーチマシンの開発，1991年情報学シンポジウム予稿集，pp.97-106 (1991)

[9] 佐藤：圧縮ファイルを用いた文字列検索の高速化，信学会論文誌Ｄ－Ｉ，Vol.J73-D-I, No.4, pp451-452 (April 1990)

[10] 山本，梅村，小長谷，横田：文字列処理とアーキテクチャ，情報処理,Vol.23, No.8, pp.719-729 (Aug. 1982)

[11]Hollaar, L.A.: Text Retrieval Computers, Computer, Vol.12, No.3, pp.40-50 (Mar.1979)

[12]高橋：テキスト検索プロセッサ，電子情報通信学会，コロナ社 (1991)

[13]Mulkhopadhyay, A.: Hardware Algorithms for Nonnumeric Computation, IEEE Trans. Comp., Vol. C-28, No.6 (Jun. 1979)

[14]Foster, M.J. and H.T. Kung: The Design of Special Purpose Chips, IEEE Computer, Vol.13, No.1, pp.26-40 (Jan.1980)

[15]高橋，永井，山田：テキストデータベースの学習型検索方式，信学技報，DE88-3 (May 1988)

[16]速水，井上：サーチプロセッサの設計と評価，データベースシステム研究会報告，51-2 (1986)

[17]伊藤，他：文書検索システム「検蔵君」（２）－－文字列照合ＬＳＩについて－－，第43回情処全大予稿，PP6-63～64 (1991)

[18]Haskin, R.: Hardware for Searching Very Large Text Databases, SIG IR, Vol. 15,No. 2, pp.49-56 (Mar. 1980)

[19]菊地，宮井：ＩＳＳＰを用いたテキスト検索システムの試作，第３５回情処全大，pp.1285-1286 (1987)

[20]菊地，杉本，辻澤：高速直接検索システム，1990年度信学会秋期全大，D-175 (1990)