ストリームデータプロセサ　ＳＤＰ（２）
パターンマッチング方式

3N-10

伊藤 正雄　早川 佳宏　田村 登　上田 謙一
松下電器産業株式会社　東京研究所

１．はじめに

本稿では、大量のデータベースから検索、更新等の処理を高速に行うストリームデータプロセサＳＤＰのパターンマッチング方式とハードウェア構成について報告する。

パターンマッチングのハードウェアに適したアルゴリズムとしては有限状態オートマトン方式[1,2]、連想メモリ方式[3]、セルラアレイ方式等あるが、本方式はストリームデータに適し、かつ柔軟なパターンマッチングが可能な有限状態オートマトン方式を用いる。有限状態オートマトン方式は、文字列を有限の入力文字と状態の２次元からなる遷移関数で表現し、入力文字で遷移関数を参照し状態を変化させることにより検索処理を実現するものである。有限状態オートマトンの利点は、正規表現を用いた複雑なパターンの検索が可能なことと、遷移関数をメモリで実現するとハードウェアが簡単で済むことである。しかし入力文字の種類が多いと遷移関数を格納する領域が非常に大きくなり、かつ前処理時間がかかるという欠点がある。

ＳＤＰは今後１チップ化を考えており、有限状態オートマトンをそのままインプリメントしたのでは、膨大な作業領域がボトルネックになると考えられる。よって本手法では、作業領域を縮小させ、かつ有限状態オートマトンの特徴を損なわない方式を考案し、インプリメントを行った。

２．ＳＤＰの検索機能

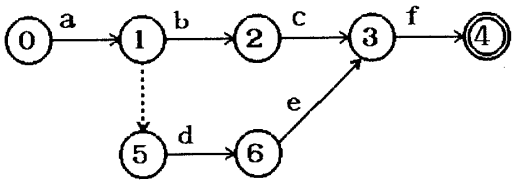
ＳＤＰの検索機能としては自然語検索ができるような柔軟な検索機能を第一目標とし、ＵＮＩＸの１ｅｘと同等の正規表現を用いた検索を実現した。またデータベースをレコードの集合、レコードをフィールドの集合とした場合、レコード全体に対して、また指定したフィールドに対して検索を可能にした。

ＳＤＰの主な検索機能を以下に示す。

・フィールド指定検索
・正規表現を含む検索
・オーダ検索
・論理演算子（ＡＮＤ，ＯＲ）を用いた複合条件検索

３．パターンマッチング方式

ＳＤＰのパターンマッチング方式は先に述べたように、有限状態オートマトン方式を用いている。有限状態オートマトン方式で有名なものとしてＡＣ法[4]がある。ＡＣ法は、goto、failure、outputの３種類の関数を用いて検索を行うものであり、データを一回走査するだけで、複数個のパターンを検索できることが特徴である。しかしＡＣ法でもgoto関数という２次元の遷移関数を用いているため膨大な作業領域が必要である。本手法では、そのような２次元の関数は用いない。その代わり、比較が失敗した場合に遷移する状態を示すfailuer関数と、比較が成功した場合に遷移する状態を示すsuccess関数と、検索パターンを格納するpattern関数と、検索の成功、失敗を示すoutput関数を用いる。図１に遷移図の例を示す。実線がsuccess関数、点線がfailuer関数、実線の上の文字がpattern関数、二重円が最終状態を示す。



ａ（ｂｃ｜ｄｅ）ｆ
図１　遷移図の例

以上の関数だけでは、可変長文字列が含まれる場合、文字範囲を指定するような検索の場合、効率が悪くなる欠点がある。そこで表１に示すように以下の関数を加える。

・**input_control**関数
比較が成功、失敗した場合に入力データのポインタを１つ進める、または進めないことにより、柔軟な比較操作を可能にするものである。

・**compare**関数
数値、文字の大小比較をする場合に、比較の種類（＝、＞、＜、≧、≦、≠）を指定する。

また、特定のフィールドに対して検索を行うために、フィールドの設定を示す**field_set_flag**、フィールド番号を指定する**field_number**、フィールドのデリミタを検出した場合に遷移する状態を示す**link**関数、比較するbyte幅を柔軟に変更できるように、**byte_width**を用いる。

以上挙げた関数（まとめて検索テーブルと呼ぶ）は全て１次元の配列であり、かつbit数は大きくないためコンパクトな作業領域で済む。表１にａ（ｂ｜ｃ）ｄを条件とした検索テーブルの例を示す。表１で**BW**, **FN**は**FS**が１の場合に有効になり、**OP**, **CO**, **IC**は**FS**が０の場合に有効になる。

４．ハードウェア構成

図２にＳＤＰパターンマッチ部（ＰＭＥ）のハードウェア構成を示す。検索するデータベースは入力時にレコード単位でバッファリングされる。**RECORD_BUFFER**は入力、

Stream Data Processor SDP (2)
The Method of Pattern Matching
Masao Ito, Yoshihiro Hayakawa, Noboru Tamura, Kenichi Ueda
Tokyo Research Laboratory Matushita Electric Industrial Co.,Ltd.

表1　検索テーブルの例

	63	32	24	16	8	6	3	1	0
状態	PA	FA	SU	LI	BW	FN			FS
					OP	CO	IC		
0	−	−	−	−	01	1			1
1	a	6	2	6	00	0	0		0
2	b	3	4	6	00	0	0		0
3	c	6	4	6	00	0	0		0
4	d	6	5	6	00	0	0		0
5	−	−	−	−	01	−	−		0
6	−	−	−	−	10	−	−		0

PA :pattern　　OP :output
FA :fail　　FN :field_number
SU :success　　CO :compare
LI :link　　IC :input_control
BW :byte_width　　FS :field_set_flag

パターンマッチ、出力のそれぞれがパイプラインで処理できるように3バッファ方式を採っている。また入力時に、フィールド単位の先頭アドレスを**RECORD_BUFFER**に格納することにより、検索時に指定したフィールドを直接読み出すことを可能にした。

ＰＭＥのハードウエア構成を実行手順を示しながら説明する。前処理として検索テーブルを**PATTERN_BUFFER**に転送する。次にＰＭＥの起動がコントローラからかかると**PATTERN_BUFFER**の先頭から読み出す。**field_set_flag**が有効の場合は、**field_number**でフィールドの先頭アドレスを得、それを用いて更に**RECORD_BUFFER**からデータを読み取る。次の状態で**PATTERN_BUFFER**のパターンと**RECORD_BUFFER**から読み取ったデータと比較を行う。本処理は、３２bit単位で比較を行うことが可能な完全一致検索と、文字コードのbit数でしか比較ができない可変長文字列等の正規表現を用いた検索を同一のハードウェアで行っている。これを実現するために8bit単位で比較を行う場合には、**BARREL_SHIFTER**を用いて**PATTERN_BUFFER**のパターンと同じbit位置にシフトを行なう。比較器の結果は**ADDRESS_CONTROLLER**に入力され、結果により**success**、**fail**、**link**が示す状態に遷移する。

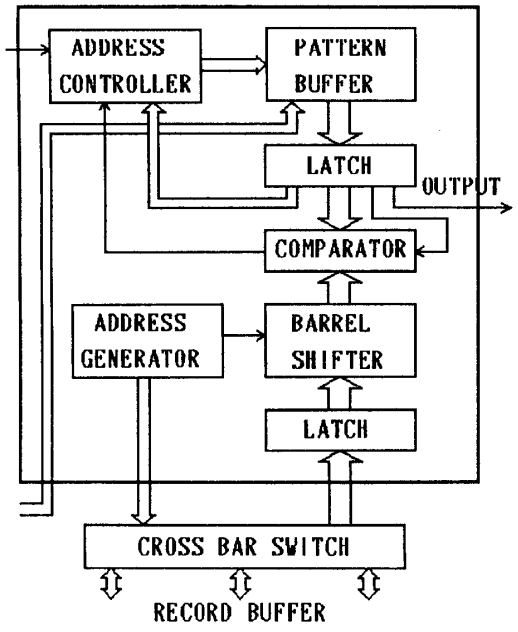


図2　ＰＭＥのハードウェア構成

5．性能予測

ここではＰＭＥの性能予測について述べる。ＳＤＰは、フィールド単位の先頭アドレスをインデックスとして持っているので、複数フィールドで１フィールドに対する検索は速いと考えられる。ここでは全文テキストデータのように全データから検索する場合について評価する。またＳＤＰは主記憶装置として大容量ＲＡＭを対象としているので、２次記憶系についてはここでは考えないものとする。比較する対象としては、汎用ＣＰＵを用いて有限状態オートマトン方式で検索する場合とする。以下に性能予測環境を示す。

・ＣＰＵ　　６８０２０、１６Ｍｈｚ、約１ＭＩＰＳ
・メモリアクセス時間（アドレス変換、バスの遅れを含む）　　４５０ｎｓ
・使用データデース　２Ｍｂｙｔｅ

以下に汎用ＣＰＵの処理時間Ｔｃを示す。

Ｔｃ＝Ｄｓ×Ｂｓ÷Ｍｉ
　　Ｄｓはデータベースサイズ
　　Ｂｓは１ｂｙｔｅ処理あたりのステップ数
　　ＭｉはＣＰＵのＭＩＰＳ値

Ｂｓを約１５とするとＴｃは３０秒となる。

次にＳＤＰの処理時間Ｔｓを示す。

Ｔｓ＝Ｄｓ×Ｔｂ×α
　　Ｔｂは１ｂｙｔｅあたりの処理時間
　　αは属性読み出しとフィールド設定時間

ＴｂはＰＭＥのサイクル時間で２００ｎｓ、αを約１．２５とするとＴｓは０．５秒となり汎用ＣＰＵより２桁処理速度が速くなることが予想される。

また作業領域の観点から見ると、8bitの文字で最大状態数２５６では本方式は2Kbyteで済むが、有限状態オートマトンでは６４Kbyte必要となるため、３２分の１の領域で済む。

6．おわりに

ストリームデータプロセッサのパターンマッチング部として、ＡＣ法のgoto関数を用いず、failuer関数を用いて作業領域を小さくし、かつ複雑な検索を可能にした検索方式とハードウェア構成について述べた。今後、実際のアプリケーションで評価を行う予定である。

【参考文献】

[1]R. W. Floyd, J. D. Ullman:"The Complilation of Regular Expressions into Integrated Circuits", Journal of ACM, Vol. 29, No. 3, July(1982)
[2]井上他:"情報提供サービスに適用可能な超大規模リレーショナル・データベースマシン",データベースシステム,47-5(1985)
[3]Takahashi, k. : "A New String Search Hardware Architecture for VLSI", Proc 13th ISCA, pp. 20-27, June, (1986)
[4]Aho, A. V. , M. J. Corasick: "Efficient String Matching : An Aid to Bibliographic Search" , CACM Vol. 18, No. 6, (1975)