情報検索システムにおける検索結果の矛盾回避手法

1H-3

石黒　正典　　青木　富夫
(ＮＴＴ電気通信研究所)

１．はじめに

オンライン情報検索システムにおける、情報の検索手順は以下のように表せる。

・条件式による検索対象の絞り込み（事前検索）

↓

・絞り込んだ対象の情報の実体（データベースのレコード）を得る（確定検索）

情報検索システムにおけるデータベースのインデックスは図１に示すように、索引語（キー）～レコードへのポインタ群、という構成であり、事前検索時における絞り込み情報は一般にレコードへのポインタを用いて管理している。このポインタ群を管理するファイルを、部分集合ファイルと呼ぶ。　一般に事前検索と確定検索の間には時間間隔があり、この間にオンライン更新が介在すると、事前検索で取得したポインタと確定検索で得るレコードとの対応関係が壊れる場合がある。

本稿では、オンライン更新を実現する情報検索システムにおいて上記のような検索結果の矛盾を効率よく回避する手法について述べる。

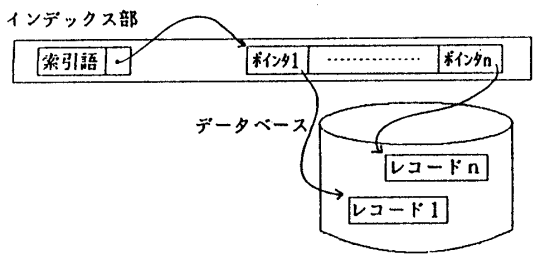


図１　検索システムのインデックス構成

２．オンライン更新が介在する場合の問題点

オンライン更新の介在する情報検索システムにおける処理の流れは概ね図２のようになる。

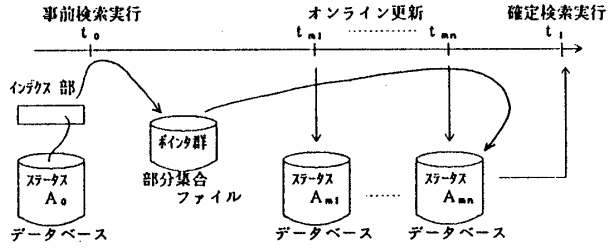


図２　オンライン更新を実現する検索システムにおける処理の流れ

オンライン更新には、レコードの追加／レコードの修正／レコードの削除、の３種類の処理がある。事前検索実行時に、ポインタ群だけで対象情報を管理する場合、図３のようなオンライン更新による影響が発生する。

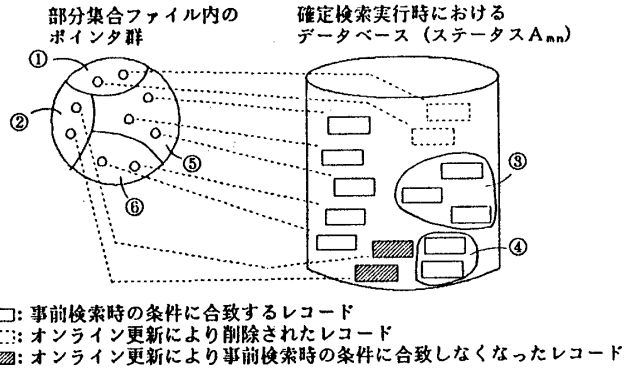


図３　オンライン更新による部分集合ファイル／データベースへの影響

①：削除処理により対応するレコードが無くなったポインタ群

②：修正処理により事前検索時の条件に合致しなくなったポインタ群

③：追加処理により新たに事前検索時の条件に合致するようになったレコード群

④：修正処理により新たに事前検索時の条件に合致するようになったレコード群

⑤：修正処理を受けたが事前検索時の条件に合致している、或は、削除後の追加処理により再度事前検索時の条件に合致するようになったポインタ群

⑥：オンライン更新の対象とはならなかったポインタ群

上記の①～④がオンライン更新が介在する場合に検索結果に生じる矛盾である。

３．矛盾回避の方針

検索結果に生じる図３の①～④の矛盾を完全に無くすには、一人のユーザが事前検索～確定検索を終了するまでデータベースをリードロックすればよいが、ロック期間中オンライン更新が不可になってしまうという問題がある。ロックをせずに効率よく矛盾を回避する方法について提案する。確定検索の対象となる個々のレコードについて事前検索時の条件式を再実行することで矛盾のチェックを行うことを基本方針とする。矛盾のチェック方法を表１に示す。但し、①の矛盾については確定検索時にポインタに対応するレコードが存在しないことによりチェックできる。

事前検索実行時以降に追加／修正されたレコード群の中から条件式再実行の対象となるレコードを抽出する方法としては、(1)オンライン更新の対象となったレコードを時刻でグループ化して管理し、事前検索実行時の時刻以降のグループのレコードを条件式再実行の対象とする、(2)データベース、部分集合ファイル等を単位としてオンライン更新

A Method to Keep Consistency of Retrieval Results in Information Retrieval Systems

MASANORI ISHIGURO , TOMIO AOKI

NTT Electrical Communications Laboratories

表1　検索結果の矛盾チェック方法

矛盾	チェック方法	条件式再実行の対象数
②	・条件式の再実行	部分集合ファイル内の確定検索対象となるポインタに該当するレコードの件数
③	・追加レコードの管理 ・条件式の再実行	追加されたレコードの件数
④	・修正レコードの管理 ・条件式の再実行	修正されたレコードの件数

の対象となったレコードを管理し、事前検索時の対象範囲に属するレコードを条件式再実行の対象とする、等が考えられる。本システムにおいては、部分集合ファイル作成時におけるポインタとレコードとの対応に着目し、オンライン更新により対応関係が壊されたもの（図3の①、②）についてのチェックは行うが、新たに事前検索の条件式に合致するようになったレコード（図3の③、④）についてはチェックの対象としないことをサービスの前提条件とした。

4．実現方式の検討

4.1　タイムスタンプ方式

確定検索の対象となった部分集合ファイル内のポインタに該当するレコードが事前検索の条件を満たすか否かのチェックには、事前検索時の条件式の保存とその再実行が必要である。この際、確定検索の対象レコードに対して無条件に条件式の再実行を行うことは、チェック不要のレコード（図3の⑤、⑥）も存在し得るゆえ処理の無駄が生じる。

このため、本システムでは以下のタイムスタンプを用いた条件式再実行の要／不要を判定する方式を実現した。

「事前検索において部分集合ファイルが作成された時点の時刻（t_0）と、確定検索実行時の時刻（t_1）に最も近いオンライン更新時刻（t_{mn}）からt_0とt_{mn}の比較を行い、$t_0 < t_{mn}$であれば確定検索の対象となったポインタに該当するレコードについて条件式再実行によるチェックを行い、$t_0 > t_{mn}$であればチェックは行わない。」

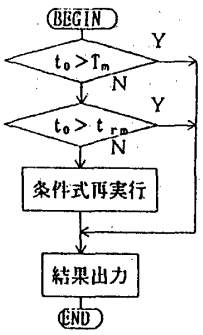
4.2　タイムスタンプ方式の実現方法

タイマの種類としては以下が考えられる。

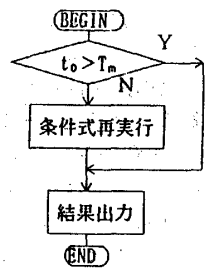
①部分集合ファイル作成時刻（t_0）：部分集合単位に管理
②レコード更新時刻（t_{rm}）：レコード単位に管理
③データベース更新時刻（T_m）：システム一意に管理

タイマを用いたチェックの方式として以下の3案がある。

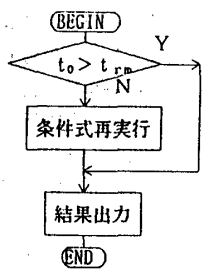
（案1）



（案2）



（案3）



$t_0 > t_{rm}$の判定はレコード単位に行わねばならないが、$t_0 > T_m$の判定はレコードをリードする必要はなくＩ／Ｏが不要となる。従って、（案2）或は（案1）の方が（案3）よりも効率が良いが、T_mをシステム一意として管理した場合$t_0 > T_m$となるのは希である。本システムでは、データベースを構成するファイルの全ての部分が事前検索の対象となりうる作りであり、T_mをシステム一意として管理せざるを得ないが、事前検索時に検索対象範囲に制限を設けるようなシステムでは、その範囲を単位としてT_mを管理すれば、$t_0 > T_m$となる場合も比較的ありうる。

事前検索～確定検索の時間幅が短く、その間にオンライン更新が発生しない場合が多いようなサービスも考慮し、T_mを管理する／しないをオプションとして指定できる作りとし、（案1）、（案3）を実現できるようにした。

5．評価

タイムスタンプ方式を実現することにより、オンライン更新の伴わないシステムに比較してどの程度ＤＳ（Dynamic Steps：アセンブライメージでの走行ステップ数）が増加するかを評価する。ただし、$t_0 > T_m$の判定は行わない（案3）を前提とする。

ＤＳの増加は以下の２つの処理に起因する。

①部分集合ファイル作成時刻とレコード更新時刻との比較
②条件式の再実行

時刻の比較時にデータベースからレコードをリードする処理が必要となるが、オンライン更新のないシステムにおいても、確定検索対象レコードのリードを行うため本方式の実現に伴うＤＳ増加としては扱わない。

・部分集合ファイル中、確定検索対象となるポインタ数：n
・レコード１件当りの時刻比較に要するＤＳ：D_t
・時刻比較の結果、条件式再実行となるレコード件数：m
・レコード１件当りの条件式再実行に要するＤＳ：D_c

とすれば、ＤＳの増加分は以下の式で求められる。

$$n * D_t + m * D_c$$

D_tの値は一意に決まるが、D_cの値は条件式の数、条件式の複雑さ等により以下のような関数で表される。

$D_c = f$（条件式構成要素、演算子の種類、条件式の数）

ここで、n，mに着目する。文献サービスを例とした場合、レコード件数は１００万件近い。検索事例を調査すると、nの値は高々１００件である。オンライン更新が多発しても、１００万件のレコードの中から確定検索の対象となる１００件のレコードが更新対象となることは希と考えられる（オンライン更新が全レコードに対して一様に行われることを前提とする）。mの値は、０となる場合がほとんどであると仮定すれば、ＤＳの増加分は、おおよそ

$$100 * D_t$$

となる。ここで、D_tは高々１０stepsであり、オンライン更新を行わないシステムと比較して、約１KstepsのＤＳ増となりユーザへの応答時間にほとんど影響を与えず、従来の検索システムと遜色ないサービス性を保障できる。

6．結言

オンライン更新を実現する情報検索システムにおいてのタイムスタンプ方式を用いた効果的な検索結果の矛盾回避方法について述べた。本方式の実現により、オンライン更新を実現しない従来の情報検索システムと同程度の情報の正確さ、検索の応答性を保障することができる。