

クライアント・サーバーにおける検索キャッシュの提案

長田弘康

鉄道総合技術研究：OA開発推進部

OA等のアプリケーションは、データベースをアクセスするサーバーとGUIを担当するクライアントから構築されるのが通常である。GUIの進展に伴い、リストボックスに件名を表示するような簡単な検索をサーバーに投げようとする例が多くなりつつある。そこで、サーバーからの検索結果をクライアントにキャッシュする方式を考案した。サーバーにあるデータベースの変更部分は各テーブルに対する更新した版を示す番号が必要となった。クライアント側には、テーブルの版と検索に使用したコマンド(SQL文)と検索結果のキャッシュが必要となった。

本提案では、この検索キャッシュの具体的なアルゴリズムを示す。

1. はじめに

OA等のアプリケーションは、データベースをアクセスするサーバーとGUIを担当するクライアントから構成されるのが通常である。このようないわゆるクライアント・サーバーアプリケーションでは、前記したような典型的な例ばかりではなく、サーバー側にロック機能等、クライアント側に置くとアクセスが輻輳するようなプログラムを置く場合もある。または、かなりの部分をアプリケーションをサーバーに置くという考え方も成り立つ。

当所では、このようなクライアント・サーバー型のアプリケーションで日常の業務を支援する統合OAシステムを作成/運用/保守をしている。統合OAを開発した経験では、システムの汎用性を高めるために、クライアント・プログラムの中にコードを埋め込むのではなく、サーバー上にあ

るデータベースをコード表を取り出すために検索するようなことがしばしばある。たとえば、メールを出すのに所属箇所コードと所属箇所名のテーブル一覧を表示され、選択させ、次に、所属箇所コードに属する職員コードに対する職員名の検索結果一覧等を表示させ、選択されるというようなものが典型例である。

このようなほとんど変化しないデータベースも、毎回検索することは効率的ではない。しかし、クライアント・プログラムにコード表を持たせるのも保守上問題がある。別の解法としては、クライアントとサーバーが同じレベルに属するとして分散データベースとして、2フェーズコミットを実装することも考えられる。しかし、そうすると、各クライアントにもデータベースを持つということになり、かつ、クライアントも常時電源を入れて置かなければならず、また、クライアントの数を考えると実用的ではない。

そこで、サーバーからの検索結果をクライアントにキャッシュする方式を考案した。

サーバーにあるデータベースの変更部分は各テーブルに対する更新した版を示すバージョン番号を持つバージョン表が必要となった。クライアント側には、キャッシュを管理するためのファイル（キャッシュ情報ファイル）と検索結果のキャッシュファイルが必要となった。キャッシュ情報ファイルのレコードは、検索に使用したコマンド（SQL文）と、そのコマンドで参照したテーブルのバージョン番号（複数）と検索結果のキャッシュファイル名である。

2. 本方式のために必要となるデータベース等の変更点

サーバー側では、データベースに各表のバージョンを記憶するための新たな表（バージョン表）が必要となる。このバージョン表の列は表名とバージョン番号からなる。バージョン表のバージョン番号は対応する表の変更があった場合に、後で述べるように一つ増やす。

クライアントでは、検索のSQL文とその検索で参照した表のバージョン番号（複数）と検索結果の入ったキャッシュ・ファイルの名前を入れる表（キャッシュ情報ファイル）と検索結果のキャッシュ・ファイルが必要となった。ただし、この検索SQL文には副問い合わせは含めないとする。これらを図示すると図1のようになる。

クライアントのキャッシュ情報ファイルにあるバージョン番号は前に検索した時のSQL文（副問い合わせは含めない）の参照していた表のバージョン番号である。その検索結果のキャッシュファイルがクライアントにあることになる。そこで、次に検索する場合には、検索コマンド（SQL文）

をキャッシュ情報ファイルから検索し、同じものが見つかった場合には、さらにクライアントにある検索キャッシュファイルの有効性をサーバーにバージョン番号を問い合わせることにより調べる。もし、サーバー側のテーブルに変更がない場合には、クライアントにある検索結果のキャッシュファイルを利用する。さもなくば、サーバー上でSQL文を実行して結果をクライアントに持ち帰りバージョン番号と検索結果のキャッシュファイルを更新する。これが基本的なアイデアである。

サーバー	
バージョン表	
表名	バージョン番号

クライアント		
キャッシュ情報ファイル		
SQL文	バージョン番号	キャッシュファイル名
	表1 表2, .	
検索キャッシュファイル		
....		

図1 データベース等の変更点

3. アルゴリズム

クライアント・サーバーのSQLコマンド実行は、次のような手順で行なう。

前に述べたように本提案では副問い合わせがSQL文に含まれるかどうか重要な因

子となる。そこで副問合わせを含まないSQL検索文をSQL基本検索コマンドと呼ぶことにする。副問合わせを含む場合をSQL複合検索コマンドと呼ぶことにする。

3. 1 SQL基本検索コマンド

SQL基本検索コマンドをクライアントのアプリケーションから発行する場合は、まずクライアント上のキャッシュ情報ファイルにあるSQL文、各テーブルのバージョン番号、キャッシュファイル名のレコードのSQL文と比較を行い、一致するものがある場合とない場合にわかれる。

場合1：SQL基本検索コマンドに対応するキャッシュファイルがある場合

(1) サーバーから、SQLコマンドで参照されているテーブルのバージョン番号をバージョン番号表より全て得る。

(2) キャッシュ情報ファイルにある各テーブルのバージョン番号を得る。

(3) もし、キャッシュ情報ファイル上のバージョン番号が全て、サーバー上の番号よりも新しいか等しいならばキャッシュファイル上のデータを検索結果とする。さもなければ、キャッシュ無しSQLコマンド検索と同じ。

場合2：SQL基本検索コマンドに対応するキャッシュがない場合（キャッシュ無しSQLコマンド検索）

(1) サーバー上でSQL基本検索コマンドを実行し、検索結果を得る。

(2) サーバーから、上記SQLのテーブルに関するバージョン番号を得る。

(3) クライアント上のキャッシュファイ

ルに検索に使用したSQLコマンド、バージョン番号、検索結果を置く。

3. 2 SQL複合検索コマンド

(1) サーバー上でSQL検索コマンドを実行し、検索結果を得る。

3. 3 SQL更新コマンド

(1) サーバー上でSQL更新コマンドを実行する。

(2) 上記SQLコマンドで更新したテーブルのバージョン番号をサーバー上で一つ上げ、蓄える。

3. 4 SQL挿入コマンド

(1) サーバー上でSQL挿入コマンドを実行する。

(2) 上記SQLコマンドで挿入したテーブルのバージョン番号をサーバー上で一つ上げ、蓄える。

3. 5 SQL削除コマンド

(1) サーバー上でSQL削除コマンドを実行する。

(2) 上記SQLコマンドで挿入したテーブルのバージョン番号をサーバー上で一つ上げ、蓄積する。

3. 6 SQLテーブル作成コマンド

(1) サーバー上でSQLテーブル作成コマンドを実行する。

(2) 上記SQLコマンドで作成したテーブル名称に対応するテーブルのレコードを作成し、そのバージョン番号をサーバー上

で0にする。

3. 7 SQLテーブル削除コマンド

(1) サーバー上でSQLテーブル削除コマンドを実行する。

(2) 上記SQLコマンドで削除したテーブル名称に対応するテーブルのレコードを消去する(サーバー上)

3. 8 SQLテーブル変更コマンド

(1) サーバー上でSQLテーブル変更コマンドを実行する。

(2) 上記SQLコマンドに対応するテーブルのバージョン番号をサーバー上で1上げる。

3. 9 サーバーでのSQLコマンド実行

上記していたものは全てクライアントからのSQLコマンド実行であるが、サーバーでのSQLコマンド実行では、それが単独のコマンドであるかプログラムに埋めこまれたコマンドであるかに関係なく、更新、削除、挿入などの表を変更する場合は、表に対応したバージョン表のバージョン番号を一つあげることが必要である。

4. おわりに

本方式はまだ提案にすぎなく、2、3の問題を解決しなければならない。考えられる問題点としては以下のものがある。

検索結果のキャッシュファイルのセキュリティ上の問題がある。クライアントは一般にパーソナルコンピュータなので、単な

るパスワードによるセキュリティの確保は難しく、暗号化等の対策が必要である。また、クライアントのキャッシュファイルの内容が故意に変更された場合、誤動作する問題がある。これも上記のような暗号化等の対策が必要である。サーバー上のバージョン表にロックが集中する問題がある。これは、シミュレーションや試作により実験して本提案の効果とこの欠点のトレードオフを調べる必要がある。

なお、本研究所の統合OAシステムのプログラムを分析すると本方式は十分有効であろうと予測される。

謝辞 本研究のヒントを頂いた鉄道総合技術研究所 輸送システム開発推進部 八賀明主幹に深謝します。