

移動分散データベースにおける質問処理

上林彌彦 早瀬道芳

京都大学工学部 岡山県立大学

本稿では移動通信環境における質問処理の問題を扱う。この場合には移動端末の電力の関係から、通信コストに非対称性があるため、分散データベースの質問処理問題としてもこのために新しい問題が生じている。移動端末のうち移動中のものについては、できる限り送信データ量を減らし電力を使わないような質問処理方式が必要である。基本的な考えは、分散データベースで一般的な準結合とデータ圧縮を組み合わせるものである。準結合時に、データ量の多い送信は、できる限り基地局やオフィスに固定されて電源に問題のない移動端末を使うようにする。またモデルとして、移動端末のデータはディスククラッシュが起りやすいことから、バックアップを基地局で行っている場合についても検討している。バックアップデータを利用することにより移動端末からの送信量を大幅に減らせる可能性がある。

Query Processing for Distributed Databases with Mobile Sites

Yahiko Kambayashi

Kyoto University

Michiyoshi Hayase

Okayama Prefectural University

In this paper we will discuss query processing problems for distributed databases with mobile sites. Due to the power saving requirement for moving sites, there are asymmetric cost functions for communication and thus we have to solve a new problem. As data transmission is the major factor to consume power, reduction of the amount of transmission is very important. We use semi-join methods with data compression technique for power reduction. Furthermore, use of back-up data spread at fixed sites is also discussed.

1. まえがき

分散データベースの質問処理では、当初通信コストが最大のコスト要因となっているため、通信量の最小化が重要であり、そのための結合の手法が開発された。しかし、高速通信網の実際化により通信コストと処理コストの兼ね合いを考えるものが主流となっていた。本稿では移動通信環境における質問処理の問題を扱う。この場合には移動端末の電力の関係から、通信コストに非対称性があり、このために新しい問題が生じている。

最新のシステムの高性能化と小型化、さらに通信技術の進歩によって、移動端末を含めた分散データベースが実用化しつつある。移動端末がオフィスに置かれている場合には電力の問題がないが、本当に移動中の場合はできる限り消費電力を減らす必要が生じる。一方移動端末による受信は送信に比べるとはるかに少ない電力ですむ。Imielinskiらは受信時間を最小にするような情報伝送を考えているが、実際に問題となるのは送信時間である。本稿ではそのような背景から、移動端末のうち移動中のものについてはできる限り電力を使わないような質問処理方式について検討する。

基本的な考えは上林が文献 [1] [2] で示した、ビットベクトルによるデータ圧縮を用いるもので、特に放送機能のあるローカルエリアネットワークで有効であった。(ローカルエリアネットワーク方式は後に滝沢によって独立に見出された [4])。この方式を用いると本質問の最適化に対してダイナミックプログラミングを使わなくてもすむ。さらに我々はこの手法を用いて巡回質問等の一般的な質問に対して効率的な処理のできることを示した [5] [6] [7] [8]。

以上の応用では、通信コストの最小化を第一の目標とし、次に並列性等を考えて処理時間の最小化を扱うものであった。ここでのモデルとの最大の違いは通信コストの非対称性である。すなわち、基地局やオフィスにある移動端末からの送信コストに比べると、外部電源のつながっていない移動端末の送信コストははるかに大きい。このため、できる限り後者からの送信量を減少させるものとする。移動端末の電力に余裕がなければ拒否されることも起こりうる。

またモデルとして、移動端末のデータはディスククラッシュが起こりやすいことから、バックアップを基地局で行っている場合についても検討している。バックアップデータはすこし前の時点のデータではあるが、これを利用することにより移動端末からの送信量を大幅に減らせる可能性がある。

2. 基本的定義

属性集合 $\{A_1, A_2, \dots, A_n\}$ 上の関係を $R(A_1, A_2, \dots, A_n)$ または R で表わす。 R の各要素 (a_1, \dots, a_n) は組と呼ばれ、各 a_i は属性 A_i の定義域 $\text{dom}(A_i)$ の要素である。 R の関係スキーマは $R = \{A_1, A_2, \dots, A_n\}$ で定義される。 R の属性 $A_{i,k}$ は $R.A_{i,k}$ で表わされる。

データベースにおける質問を簡単のため関係代数質問とする。再帰質問は扱っていないが、同じ関係集合に対して結合処理を繰り返し適用するために、同様の手法が利用できる。関係代数質問は、射影 (R の属性集合 X 上への射影は $R[X]$)、選択、結合 (R_i と R_j の結合は $R_i \times R_j$ で示す) および集合演算によって表現される。分散データベースでは、結合演算のみが局間にまたがる演算であるために、結合演算のコストが最も高くなるため、質問処理を行う際に不要な属性の射影による除去、不要な組集合による除去といった前処理を行った関係のみを考えて、結合演算の最適化だけを問題とする。さらに議論を単純化するため、結合の中でも本質的な自然結合のみを扱うものとする。

等結合質問は、 $R_i.A_{i,k} = R_j.A_{j,k}$ の間の結合の集合で表現される。ここで $R_i.A_{i,k} = R_j.A_{j,k}$ (すなわち

属性名が同じ) という条件をすべての結合表現式が満足するとき、準自然結合質問と呼ぶ。この場合は結合に使われた属性のうち片方は内容が全く同じであるため省かれる。すべての $A_{ijk} = A_{jki}$ に対して $R_i A_{ijk} = R_j A_{jki}$ の形の結合表現式が存在するとき、すなわち同じ名前の属性のすべてを用いる準自然結合質問を、自然結合質問と呼ぶ。任意の等結合質問は、名前の付け替えで自然結合質問に変換できる。準自然結合質問は自然結合質問の等価変換のために必要である。

準自然結合質問 q に対する質問グラフは、 $G(V_q, E_q, L_q)$ で表現される。ここで V_q は節点集合で、各節点は質問 q が用いる各関係に対応している。 $R_i.A = R_j.A$ という結合表現式が存在すれば、 R_i に対応する節点と R_j に対応する節点との間が A というラベルのついた枝で結ばれる。このような結合表現式が複数個あると、これらのすべての属性名の和集合をラベルとする。

$R_1.A = R_2.A, R_2.A = R_3.A, R_1.A = R_3.A$ という3つの枝のある場合、最後の枝は他の2つの枝から推移的に導けるために除いてよい。このような変換を質問グラフの等価変換と呼ぶ。質問はそれを表現する質問グラフを等価変換して木状にできるとき、そのときに限り木質問と呼ぶ。どのように変換しても木構造とならないものを巡回質問と呼ぶ。木質問の部分クラスですべての関係が同一の属性(集合)で結合されるものを単純質問、図1(b)に示すような1つの道だけからなるものを鎖質問と呼ぶ。

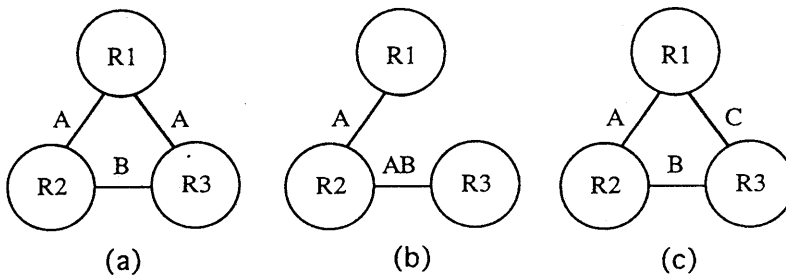


図1 木質問と巡回質問

【例1】 図1(a)に示すように次の結合表現集合は巡回質問にみえる。

$$R_1.A = R_2.A, R_2.B = R_3.B, R_1.A = R_3.A$$

しかし、 $R_1.A = R_2.A, R_1.A = R_3.A$ は先に述べたように、 $R_1.A = R_2.A, R_2.A = R_3.A$ および $R_1.A = R_3.A$ のうちの任意の2つの集合と等価であるため、次の結合集合に変換できる。

$$R_1.A = R_2.A, R_2.B = R_3.B, R_2.A = R_3.A$$

これは図1(b)に示すように木質問となる。

次の質問はどのようにしても木質問にならない巡回質問である。

$$R_1.A = R_2.A, R_2.B = R_3.B, R_1.C = R_3.C$$

この質問グラフを図1(c)に示す。

R_i の R_j による準結合を $R_{ij} \times R_j$ で示す。

$$R_{ij} \times R_j = (R_{ij} \times R_j) \quad [R_i] = R_{ij} \times R_j \quad [R_i \cap R_j]$$

定義は2番目に示すように結合結果を R_i の属性集合に射影したものとなる。実現法は3番目に示すように、関係 R_j を結合属性集合 $[R_i \cap R_j]$ に射影したものを R_i と結合するもので、この2つの関係が別の局にあれば、前者を送信することにより通信量を削減できる。準結合操作は結合操作よりも処理コストが低

く、実際に結合を行う前の前処理として優れている。また木質問に対しては、すべての部分解が準結合によって求められる。

ここで、 R_1, R_2, \dots, R_n に対する自然結合質問を

$$R = R_{11} \times R_{21} \times \dots \times R_n$$

とすると、 R_i ($i = 1, \dots, n$)に対する部分解は $R[R_i]$ である。

【準結合を用いた木質問処理】 [12]

(上昇階段)

- (1) 質問グラフの葉のほうから根のほうへ向かって順次準結合が実行される。 R_i に直接つながり葉の方面にある関係を $R_{j1}, R_{j2}, \dots, R_{jp}$ とし、それらに対してはすでに準結合が実行されているとする。 $R_i \times R_{jk}$ ($k = 1, \dots, p$)を実行する。この場合に R_{jk} の中の $R_{jk} \cap R_i$ の値の組合せが R_i のある局へ送られる。
- (2) 上記(1)を繰り返す。
- (3) 木の根に至ると、根の関係に対する部分解が得られる。

(下降段階)

- (1) 木の根の方向から葉の方向へ順次準結合を適用し部分解を求める。上昇階段の(1)と同じ記号を用いることにする。 R_i に関する部分解が求まったとして、これを $R_{j1}, R_{j2}, \dots, R_{jp}$ に送る。すなわち、 R_{jk} ($k = 1, \dots, p$)を $R_{jk} \times R_i$ で置き換える。
- (2) 上記をすべての関係の部分解が求まるまで繰り返す。

分数データベースの局を S_1, S_2, \dots, S_p とする。 $R(S_i)$ は局 S_i に蓄えられている関係集合である。同じ関係が複数個の局にあることを許しているので $R(S_i) \cap R(S_j) \neq \emptyset$ のことがある。 $(i \neq j)$ 。

3.分散処理のためのデータ圧縮法

文献[1]に従い、分散データベース処理のためのデータ圧縮法をまとめる。結合処理の基本は、2つの局に蓄えられた2つの集合間の共通集合をとることにある。

例えば、 S_1 に〈YAMADA, TANAKA, YOSHIDA, GOTO〉があり、 S_2 に〈YAMADA, YOSHIKAWA, UEDA, OKAMOTO, YOSHIDA〉があるとすると。

【一般の方法】 S_1 の全データを S_2 に送る。 S_2 で共通集合演算を実行する。

結果の〈YAMADA, YOSHIDA〉を S_1 へ送る。

これは次のように一般化される。

- (1) S_1 と S_2 のデータのうち少ない方を相手側に送る。
- (2) その局で共通集合演算をとり他の局へ送る。

ところが、上記の例では S_1 は S_2 に送った全データをもっており、 S_2 より送り返されてくるのはその部分集合となっている。従って、 S_1 のデータをその順に、YAMADA=1, TANAKA=2, YOSHIDA=3, GPTP=4

とすると、整数値として1、3のみの符号を送り返すとよい。また、長さ4のビットベクトルを用意し(1010)とすればよい(1番目と3番目が1)。このようにすればステップ(2)におけるデータ転送はほとんど重視できる位に少なくなる。

すなわち次の2つの方法が考えられる。

- (1) ビットベクトルの利用。
- (2) データの圧縮値としての整数の利用。

ビットベクトルを用いる方法は、データ圧縮量が非常に大きい、他のデータ値との組合せを示す場合は数字の方が適していることもある。巡回質問処理のアルゴリズムでは数字を用いている[5]。

データの順序については種々のものが考えられる。

- (1) データを送った順(これは送った側の局で蓄えられている順等)
- (2) データ値の大きい順ないしは小さい順

複数属性の場合も、(1)は同じように使える。(2)は属性1の値で整列させ、その次に属性2の値で整列させ... という操作を繰り返すとよい。複数属性の場合は、単一属性ごとに処理を行ったあとその記号を使う方法も考えられる。

[例2] 以下のような組のある S_1 と S_2 を考える。

$S_1: \{ (YAMADA,50), (TANAKA,30), (YOSHIDA,30), (GOTO,50) \}$
 $S_2: \{ (YAMADA,50), (YOSHIKAWA,30), (UEDA,20), (OKAMOTO,40), (YOSHIDA,40) \}$

まず S_1 より第1成分のみを S_2 に送る。 S_2 より(1010)が送出されるのでYAMADA=1,YOSHIKA=2という記号を割り当てる。次に S_1 より第2成分の{50,30}を S_2 に送るとこれらはすべて S_2 に含まれているので(11)が送り返され、50=1,30=2という記号が割り当てられる。この2つの組合せは4通りあるので11,21,22の順とすると、 S_1 には11と22があることになり S_1 より S_2 に(1010)というベクトルを送る。 S_2 には11しかないので(10)を送り返すことになる(相手から送られた1の部分のみに対応)。この方法によれば、単一属性の実データ通信の後はずべてビットベクトルになる。

[複数属性のビットベクトル符号化]

- (1) 適当な順で送りその順を番号とする。
- (2) 属性に適当な順序を付け、その準に整列操作を適用する。
- (3) 各属性ごとに処理をして符号化を行ない、その各組み合わせをビットベクトルの各位置に対応させる(例1の方法)。

4. 質問の最適処理

準結合を用いて鎖質問や木質問を処理する場合、冗長な準結合操作を追加しても、それによって得られる処理コストの低下がその準結合処理のコストより大きければ、有効であると言える。例えば、図2に示すような木質問が与えられた場合、 R_1 のAの値集合を R_2 のある局へ送るのが基本的な準結合であるが、 R_2 の持つAの値集合の方が少ない場合、 R_2 より R_1 の方へAの値を送り、その結果のAの値の集合を R_2 へ送る方が通信コストの減ることがある。このような冗長準結合操作を考えた場合、最適解を求めるためには動的

計画法を用いる必要がある [9] [10]。しかしながら、圧縮データ量が0に近いという仮定のもとでは直接的に解を出すことが可能となる。

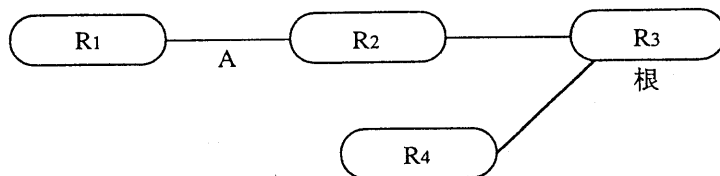


図2 冗長な準結合の必要性

ビットベクトルを用いたデータ圧縮を行なうと、実データの転送は始めの1回のみで済み、あとはその集合の部分集合を扱うためのビットベクトルの転送の繰り返しとなる。ビットベクトルの転送は実データに比べるとはるかに少ない（ n 組のデータの代わりに n ビット）ので無視できると仮定してよい。

以下では、電源に繋がれていない移動端末を省電力端末とよび、その他の固定局および電源に繋がれている移動端末と区別する。問題は次のようになる。

- (1) 省電力端末の消費電力をできる限り減らす。
- (2) 一般の移動端末が省電力端末になったり、その逆も起り得る。このため動的変化も扱えなければならない。

省電力端末に蓄えられている関係と結合される関係には次の2種がある。

- (1) 省電力端末ではない。
- (2) 省電力端末である。

また、複数個の関係と結合する場合に省電力端末からの送信はできる限り放送機能を利用して送信回数を減らすことも重要である。省電力端末から直接放送できる端末には距離の制限があるため、実際には信号を受け取った固定局が放送することとなる。

基本的な考え方を示すため、図2の例を用いる。

[図2の R_1 が省電力端末に含まれ R_2 はそうでない場合]

- (1) R_2 より結合属性のデータを R_1 に送信する。
- (2) R_1 よりは結合結果のビット列を送信する。

[図2の R_1 も R_2 も異なる省電力端末に含まれている場合]

(1) $|R_1[A]| = n_1$ 、 $|R_2[A]| = n_2$ とする。1組あたりの通信コストを C_{12} とすると $C_{12} \times \min(n_1, n_2)$ がこの準結合コストとなる。すなわち、 n_1 と n_2 の小さい方を送れば、後はビットベクトルのみでよいためである。

(2) 送信局の選択は、残っている電力と送信コストの両方から決められる。送信されたデータは固定局に蓄えられるので、下降段階の処理の一部は固定局で代行できる。

[図2の R_2 が省電力端末に繋がれていて R_1 と R_3 はそうではない場合]

- (1) R_1 より結合属性データを送る。
- (2) R_3 より結合属性データを送る。
- (3) R_2 で準結合を行なう。

- (4) R1およびR3の結合属性は複数属性としてデータ圧縮してR1およびR2に送る。
- (5) 以下の処理はR1とR3の間で直接行なう。
- (6) 下降段階でR2に最後に残るデータが決ったあと、R2から必要ならば結合属性以外のデータ値を送る。

さきに、巡回質問を属性付加によって木質問化する方法を示した。この場合に属性付加によって枝の一部が取り去られるが、どの枝を取り去るかも電力条件を考えて実行することとなる。

以上を考えると方式の概要は以下のようにまとめられる。

[省電力端末を考慮した質問処理]

- (1) 省電力端末とそうでない端末や局にある関係との結合では、省電力端末からの送信をビットベクトルになるようにする。
- (2) 省電力端末からの送信はできる限りまとめて、複数属性のビットベクトル化により、送信回数を減らす。
- (3) 省電力端末同士では、送信量と残っている電力によって送信負荷配分を決める。結合属性の値によって分割し、片方は一方がビットベクトル化をし、他方はもう一方がビットベクトル化をするという方式で、負担の比率を任意に選べる（一属性によって3つ以上の関係を結合する場合は自由度は減少する）。
- (4) 下降段階の処理はできる限り固定局が行う。これは、上昇段階での端末からの送信データを利用できるためである。

5. バックアップコピーの利用

移動局に含まれるデータは失われ易いという問題がある。このため移動局に含まれる関係のコピーが固定局で記憶されているとする。この場合、データの変更はすべて反映される必要がある。各コピーが全く同じであるとする、実データではなく、組番号を送るだけで変更の反映ができる。実際には、バックアップコピーと実データの間には差がありうるが、これは差分ファイルやその間の処理を記憶することで対処するものとする。

[コピーにデータの変更の反映のためのデータ圧縮を用いた方法]

- (1) 削除：i番目の組を削除するとき、削除記号の後にiのみ送るとよい。
- (2) 追加：i番目の組の次に新しい組を追加するときは、追加記号、i、新しい組の内容を送るとよい。
- (3) 更新：i番目の組の内容の変更は、変更記号、i、新しい内容を送るとよい。

バックアップコピーが完全に内容を反映しているとする、処理は次のようになる。

- (1) 質問処理はすべて固定局で行う。
- (2) 処理結果を移動局に送り、そのデータを処理後の結果と同一にする。

さらにコピーが多重である場合には、固定局どうしの処理についてもコストの低減をはかりうるが、ここでは省略する。

紙数の関係で詳細なアルゴリズムよりも考え方を示すにとどまったが、過去における通信量最小問題が新しい問題に少しの変形で適用できることを示した。

文献

- [1] 上林弥彦, " 圧縮型準結合による分散データベースの質問処理, " 電子通信学会技術研究報告, A L 8 1 - 5 4 , 昭和 5 6 年 9 月。
- [2] Kambayashi, Y. "Theory of Databases" in Computer Science & Technologies 1982, pp.206-226, OHM North-Holland, March 1982.
- [3] Hevner, A.R. and Yao, S.B., "Query Processing in Distributed Database Systems," IEEE Trans. Software Eng, SE-5, no.3, 1979.
- [4] 滝沢、京都大学大型計算機センター研究セミナー報告、昭和 5 7 年 3 月。
- [5] Kambayashi, Y. , Yoshikawa, M. and Yajima, S., "Query Processing for Distributed Databases Using Generalized Semi-Joins," Proc. ACM SIGMOD, pp.151-161, June 1982.
- [6] Kambayashi Y. and Yoshikawa, M. , "Query processing Utilizing Dependencies and Horizontal Decomposition, " Proc. ACM SIGMOD, pp.55-67, June 1982.
- [7] Yoshikawa, M. and Kambayashi Y., "Processing Inequality Queries Based on Generalized Semi-Joins," Proc. VLDB, pp416-428, Aug. 1984.
- [8] Kambayashi Y. , "Processing Cyclic Queries," in Query Processing in Database Systems, Springer-Verlag, 1985.
- [9] Chiu, D-M., Bernstein, P>A> and Ho, Y-C, "Optimizing Chain Queries in a Distributed Database system," Report, Harvard Univ. 1981.
- [10] Yu, C.T., Ozsoyoglu, Z.M. and Lam, X., "Optimization of Distributed Tree Queries," JCCS, vol.29, pp.409-449, 1984.
- [11] Pelagatti, G. and Schreiber, F.A., "Evaluation of Transmission Requirements in distributed Database Access," Proc. ACM SIGMOD, May 1979.
- [12] Bernstein, P.A. and Chiu, D-M, "Using Semi-Joins to Solve Relational Queries," JACM, 28, no.1, pp.25-40, Jan. 1981.
- [13] Imielinski, T., 情報処理学会データベースシステム研究会第100回記念招待講演、平成 6 年。