マルチデータベース環境における問い合わせ処理

4W－1

西澤 格　高須 淳宏　安達 淳
東京大学工学部　学術情報センター研究開発部

1 はじめに

近年のネットワーク技術の進展と情報サービスの急速な普及に伴い、異なるサイトにある複数データベースへの統合的なアクセスの要求が現実的になってきた。複数のデータベースに対する統合的な問い合わせ処理は各データベースのスキーマが異なるため、一般には困難な問題である。しかしスキーマが異なるにも関わらず、同時に問い合わせを行なう要求の発生は、対象となるデータベースがよく似た内容の情報を保持している場合である。利用者は異なるデータベースから情報を収集しようとする場合、各データベース毎に異なるコマンドを用いてアクセスし、さらにその結果をスキーマに応じて併合する必要がある。利用者の立場からはそのような処理が自動的に行われると都合がよい。

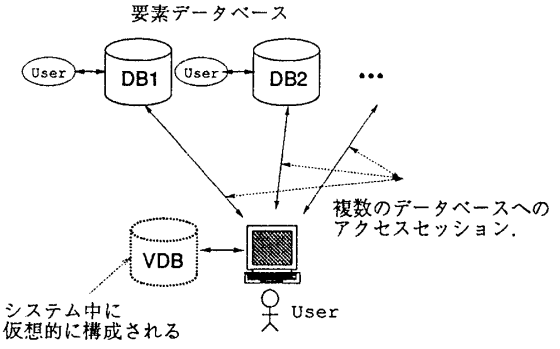

図 1: 複数データベースへの統合的アクセス

2 データベースへの統合的アクセス

2.1 想定する環境

本稿ではスキーマを属性の集合と限定した場合の分散データベースのスキーマの統合方法について述べる。各データの一貫性については各データベース上のスキーマレベルでの一貫性を考え、空値の考え方を導入することにより、この仮定の下で、論理的な観点から分散環境で正しく、かつ完全な解を得るための問い合わせ処理の方法について議論する。この実現のために本稿ではデータベースを論理式の集合とみなすアプローチ[2]を採用する。このアプローチはその厳密さから解の正しさや完全性を議論するのに都合がよい。

本稿で想定している環境は図1に示されるような異種データベース[1]環境である。各々のデータベースを要素データベース（以下 CDB）と呼ぶ。各 CDB は独立した主体によって自律的に構成され、運用されているため、そのスキーマはその構造や実体の表現に違いがある。問い合わせは仮想的に構成される仮想データベース（以下 VDB）に対して発行される。この VDB の定義は 2.2節でなされる。VDB は物理的に分散されたデータベースから構成され、VDB に対して発行された問い合わせは 2.2節で定義される仮想関数従属性（以下 VFD）を用いることによって変形され、各 CDB に対して発行される。システムは各 CDB から得られた解を併合し、利用者に示す。

2.2 諸定義

スキーマ統合における VDB の概念は直観的には物理的に分散しているデータベースを合成するというもので、そのスキーマは各 CDB のスキーマの和集合となる。VDB は実際に構成されるのではなく、仮想的なものであり、実際には問い合わせは分解され、各 CDB に対して発行される。i 番目の CDB を DB_i と表す。DB_i 上のすべての属性は VDB 上にも存在する。CDB DB_i の各タプルについて、VDB のスキーマにあって DB_i のスキーマにない属性の値に空値（Null Value）を補ったものを DB_i' とすると VDB を $VDB = \bigcup_i DB_i'$ と定義する。

本問い合わせ処理手法では各 CDB 上の関数従属性の集合から VFD を抽出する。VFD は CDB 上の関数従属性の集合から構成されるが、その定義のしかたによっては本稿で提案する問い合わせ処理において効果がなかったり、構成される VDB 上のスキーマが矛盾を含むものとなってしまう。これらを考慮して、本稿では VFD を以下の式 (1) のように定義する。

$$VFD = \{f \mid (\forall i)(attr(f) \subset U_i \Rightarrow f \in D_i^+)\} \quad (1)$$

ここで、CDB DB_i 上での属性集合を U_i、DB_i 上でのある一つの関数従属性を f、DB_i 上での関数従属性の集合を D_i、D_i によって論理的に含意される関数従属性の集合（D_i の閉包）を D_i^+、関数従属性 f の両辺に現れる属性の集合を $attr(f)$ とした。式 (1) による VFD は直観的には注目する属性が存在するすべての CDB 上で成立する関数従属性の集合である。

VDB 上での VFD を用いた問い合わせ処理法によって得られる解集合は、VFD を用いない場合よりも大きなものとなる。そこで、この大きくなった解の正しさが問題となり、VFD を適用して得られた解も論理的に正しく矛盾を含まないことを示す必要がある。ここでは紙面の都合上この証明は割愛するが、まず VDB を公理によって表現し、式 (2), (3) に示すように、問い合わせとそれに対する解も論理式で表現することによって証明を構成することができる。ただし式 (2), (3) において、T はデータベースを表現する公理、τ はドメイン、W は問い合わせの論理式、$\vdash$ は論理的帰結を表すものとする。

$$Q = \langle \vec{x}/\vec{\tau} \mid W(\vec{x}) \rangle \quad (2)$$

$$T \vdash \tau_i(c_i) \quad i = 1, 2, \cdots, n \quad and \quad T \vdash W(\vec{c}) \quad (3)$$

3 提案した手法の有効性

3.1 問い合わせ処理の例

本稿で述べた問い合わせ処理手法の有効性を例によって示す。図 2(a) に示すような二つの CDB DB1、DB2 を考え

Query Processing in Multi-Database Environment
Itaru NISHIZAWA[1], Atsuhiro TAKASU[2], Jun ADACHI[2]
[1]Faculty of Engineering, The University of Tokyo
[2]Research & Development Department, National Center for Science Information Systems

DB1

国籍	著者	タイトル
英国	Thomas	DB
英国	Smith	情報検索
米国	Robert	人工知能

DB2

著者	タイトル
Thomas	OS
Smith	信号処理
William	DB

(a) 要素データベース

VDB

タプル ID	国籍′	著者′	タイトル′
1	英国	Thomas	DB
2	英国	Smith	情報検索
3	米国	Robert	人工知能
4	ω_1	Thomas	OS
5	ω_2	Smith	信号処理
6	ω_3	William	DB

(b) 仮想データベース

図 2: 問い合わせ処理の例のためのデータベース

る。DB1 は関数従属性として “著者 → 国籍” を保持している。この時に VDB は図 2(b) のようになる。また VFD は “著者′ → 国籍′” となる。以下の議論で便利なように VDB にはタプル ID を付加してある。

ここで「著者の国籍が英国である文献のタイトルをもとめよ」という問い合わせを考える。この問い合わせの実際の処理は以下のようになる。まず、VDB 上のタプル ID が 1 および 2 に対応するタプルが DB1 から得られる。VFD を用いることにより、Thomas および Smith の国籍が英国であることが演繹され、DB2 からタプル ID が 4 および 5 に対応する解が得られる。これは VDB において空値 ω_1 および ω_2 が英国と演繹されることに相当する。この結果、解は “DB”、“情報検索”、“OS”、“信号処理” となる。もし、同じ問い合わせを DB1 および DB2 に対して個別に発行したとすると、“DB”、“情報検索” という二つの解しか得られない。

3.2 シミュレーションによる有効性の評価

本稿で述べた問い合わせ処理を用いると個々の CDB へ個別に問い合わせを発行する場合よりも大きな解集合を得ることができる。この有効性を示すためにシミュレーションを行なった。グラフの縦軸は利用者の与えた問い合わせを各データベースに個別に問い合わせを発行した結果得られる解の個数に対する、VFD を適用して得た解の個数の比を表している。また横軸の Copy Rate はデータベースの垂直分割だけでなく、水平分割に対する本手法の有効性を示すために導入されたパラメータで、直観的には各 CDB 上でのデータの重複度を表す。

実際の問い合わせ処理では各 CDB から VDB を構成することになるが、このシミュレーションでは簡単のため最初に VDB を構成し、それをいろいろなパラメータに応じて分割し、さらに VFD を用いて VDB の再構成を行なうという方法をとっている。これにより分割は VFD、タプルとなるデータの両方を保存することになり、情報は無損失となっている。なお、実際のデータベースの条件に少しでも近付けるために、実際に使用されているいくつかのデータベースのスキーマを調べ、ドメインは 2 値の値をとるものを全 30 個のうちの 1 個、残りは値の幅が 100 と 200 のものを等確率で準備した。パラメータは VDB のドメインの数を 30、分割される CDB の数を 6、VFD の数を 6, 9, 15、各 CDB の持つ属性数を 20, 25, 28、問い合わせの条件で参照する属性数を 1, 2, 3 とした。

図 3(c) より、問い合わせで参照する属性数が多くなるほど本問い合わせ手法はその効果が大きくなることがわかる。また図 3(b) より、本手法はデータベース間のデータがある程度水平分割および垂直分割されている時に効果があることがわかる。本手法では VFD はその数が多いほど問い合わせ処理には効果があることは自明であり、これは図 3(a) により確認できたが、VFD の数が 6, 9, 15 という実際的な値で通常の問い合わせ処理と比較して最高 20%程度解集合を大きくすることができた。

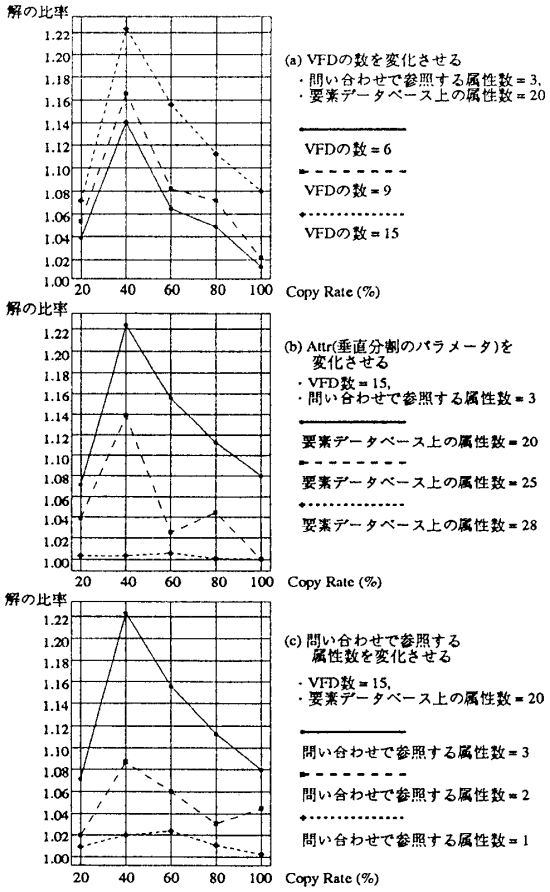

図 3: 問い合わせ手法による解の比率

4 おわりに

分散環境で複数のデータベースに対する統合的アクセスを可能とするためのスキーマ統合および問い合わせ処理手法を提案した。本問い合わせ手法では VDB を構成することにより、利用者は属性のミスマッチを考慮せずに問い合わせを記述することができ、さらにその問い合わせの意味をこわさずに複数のデータベースに対して問い合わせを発行し、論理的に正しいより大きな解集合を得ることができる。本稿では問い合わせ処理の例示とシミュレーションという二つの側面から検証を行ない、その有効性を確認することができた。

参考文献

[1] W. Litwin, L. Mark and N. Roussopoulos: “Interoperability of Multiple Autonomous Databases”, *ACM Computing Surveys*, Vol. 22, No. 3, pp. 267–293, 1990.

[2] R. Reiter: “Towards a logical reconstruction of relational database theory,” *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, Springer–Verlag, New York, 1984, pp. 191–233.