

## ATM環境内で移動するデータベースの位置管理について

原 隆浩 春本 要 塚本昌彦 西尾章治郎

大阪大学大学院 工学研究科 情報システム工学専攻

広帯域ネットワーク上では、大量のデータの転送が短時間に行なえるため、データベースの移動をデータベース技術の一つとして用いることが有効である。データベースが分散システム内を移動する環境では、特定のデータベースにアクセスするために、データベースの位置を管理する必要がある。本稿では、このような環境におけるデータベースの位置管理手法について論ずる。まず、これまでに提案されている移動体の位置管理手法や分散システムにおけるデータ項目の位置管理手法を応用した手法を考え、その後、ATMネットワークの特徴を考慮した新たな手法を提案する。そして、シミュレーションによって、これらの手法をデータベース操作のための通信所要時間の観点から比較する。そこで得られる結果は、さまざまな形態のシステムにおいて、最適な位置管理手法を同定するための指標となる。

## On Location Management for Migratory Databases in ATM Environments

Takahiro HARA Kaname HARUMOTO Masahiko TSUKAMOTO Shojiro NISHIO

Department of Information Systems Engineering, Faculty of Engineering, Osaka University

Since in broadband networks the whole contents of a database can be transferred from one site to another site in a short period, the database migration is expected to be one of the most powerful database operations in the forth coming distributed database systems. In the environment with database migration, it is necessary to develop a method to manage the location of each database for accessing migratory databases. In this paper, we first propose several database location management methods, some of which make use of the features of ATM (Asynchronous Transfer Mode) networks. Then, we compare the performance of the methods by simulation study under some system parameters, such as the frequency of query issues and database migration, and the scale of the network. As a result, we show the optimal location management method for a given system environment.

### 1 はじめに

ATM (Asynchronous Transfer Mode: 非同期転送モード) 方式などのネットワーク技術の発展により、ネットワークの帯域幅が急速に拡大し、データベース技術に大きな影響を与えている。

分散データベースシステムの研究においては、従来は伝送データ量の削減による性能向上が主要な目的であったのに対し、今後は、帯域幅の有効利用による性能向上を目的とした新たな技術の開発がより重要になる。特に、広帯域ネットワークはメッセージの伝搬遅延に関しては従来のネットワークとほぼ同じであることから、小さいサイズのメッセージを複数回交換する処理方法ではなく、ある程度大きな情報を一括して交換する手法が有効になる。

このような観点から、筆者らは、データベース

自体をネットワークを介して移動させること(データベース移動)をデータベース技術に応用し、データベース移動を利用したトランザクション処理手法を提案した[1, 4]。この手法では、データベース移動を用いてトランザクション発生サイトに必要なデータベースを移動させて集中処理することで、処理時間の短縮を図っている。データベース移動の応用は、トランザクション処理以外の分野にも、負荷分散などさまざまな目的で有効に活用できるものと考えられる。

データベースが移動するようなネットワーク環境では、データベースアクセス時にデータベースが存在するサイトまでの処理依頼メッセージの着信を保証するために、各々のデータベースの現在位置を管理する機能を実現しなければならない。従来から、移動体計算環境における移動体の位置

管理 [2, 3] や分散システム内でデータ項目の位置管理の手法などに関して、多くの研究が行われている。データベースに関しても、これらの手法をベースとして位置管理を行うことが可能である。しかし、従来の手法は、ATM などの広帯域ネットワークを想定していないため、その利点を活用するように設計されていない。

そこで、本稿では、従来の手法をベースとしたデータベース位置管理手法および ATM ネットワークの豊富な帯域幅やコネクションの一定時間の定常性といった特性を利用した新たな手法を提案する。更に、提案した手法をシミュレーションによってさまざまな環境で比較することにより、分散システムの特性に応じて最適となる手法を同定する。

## 2 データベース位置管理手法

### 2.1 ネットワークモデル

本稿では、ATM ネットワークを想定する。ATM は基本的にコネクション型の交換方式であり、コネクションには、ATM 交換機の設定によって永続的なコネクションを確立しておく PVC (Permanent Virtual Channel) と、必要時に各サイトからの要求によって設定する SVC (Switched Virtual Channel) とがある。本稿においては、PVC コネクションは仮想的な LAN の構築などの特殊な場合にのみ用いられるため、基本的な通信はすべて SVC コネクションを用いて行なうものとする。

### 2.2 位置管理手法

#### 2.2.1 移動体の位置管理手法の応用

文献 [3] では、移動体計算環境における移動体の位置管理手法として、メッセージのブロードキャストを用いた手法とデフォルトルータと呼ばれるルータに移動体の位置情報を保持させる手法を、合わせて五つ提案している。本稿では、デフォルトルータをデータベースの位置管理のためのサーバ (位置管理サーバ) に置き換えて、これらを手法をデータベースの位置管理手法に変更する。但し、文献 [3] では、各々の移動体に対してデフォルトサーバが設置されるが、本稿では、すべてのデータベースの位置情報を一つの位置管理サーバで保持しているものとする。データベースの位置

管理用に変更した手法の概要を以下に示す。

**ブロードキャスト通知 (BN) 方式:** データベースが移動する度に新たな位置情報がブロードキャストされる。従って、すべてのサイトが各データベースの位置情報を正確に把握しているため、そのデータベースが存在するサイトへと直接的に処理依頼メッセージを転送することができる。

**デフォルトフォワーディング (DF) 方式:** データベースの移動情報は、位置管理サーバのみに伝えられる。処理依頼メッセージを位置管理サーバに転送し、位置管理サーバがそのメッセージを目的のデータベースが存在するサイトへと再転送する。

**デフォルト問合せ (DQ) 方式:** DF 同様に位置管理サーバのみが各データベースの位置情報を把握している。まず、位置管理サーバにそのデータベースが存在するサイトを問い合わせ、その結果を用いて、処理依頼メッセージを目的のデータベースが存在するサイトに直接転送する。

**ブロードキャストフォワーディング (BF) 方式:** データベース移動が起こってもどのサイトにも通知されない。処理依頼メッセージを全サイトにブロードキャストし、目的のデータベースが存在するサイトはこのメッセージを受信し、他のサイトは廃棄する。

**ブロードキャスト問合せ (BQ) 方式:** データベース移動が起こってもどのサイトにも通知されない。まず、ブロードキャストを用いて全サイトに目的のデータベースが存在するサイトを問い合わせる。そのデータベースが存在するサイトから返送されるアドレス情報を用いて処理依頼メッセージの転送を行う。

ここで、すべての方式において、処理結果の転送は、問合せ発生サイトと目的のデータベースを保持するサイトとの間に確立された SVC コネクションを用いて行なうものとする。

#### 2.2.2 その他の手法の応用

文献 [3] 以外にも、移動体やデータ資源の位置管理手法がいくつか提案されている。ここでは、文献 [2] で移動体の位置管理のために提案されているポインタ (Pointer) 方式を応用した連鎖フォ

ワーディング (CF) 方式と、文献 [5] で分散システム内のデータ項目の位置管理に用いられている手法を応用した連鎖問合せ (CQ) 方式について説明する。これらの手法は、処理依頼メッセージがデータベースの移動の軌跡をたどってデータベースの現在位置まで転送される。

CF, CQ の両方式では、まず全サイトに、システム内の各データベースの初期位置を記した位置情報表 (以下では LL (Local Location) 表と呼ぶ) を持たせる。以後、LL 表は、各サイトにおいてローカルに更新されていく。その更新の規則として次のものを用いる。

更新規則：

- (i) データベース  $D_i$  が、サイト  $S_i$  からサイト  $S_j$  に移動する場合、両サイトの LL 表の  $D_i$  の現在位置を  $S_j$  に書き換える。
- (ii)  $D_i$  にアクセスするために、LL 表に従ってサイト  $S_j$  にメッセージを転送したが、既に  $D_i$  がサイト  $S_j$  に移動していることが判明した場合は、 $D_i$  の現在位置を  $S_j$  に書き換える。

各サイトの LL 表は必ずしも正確な位置情報を把握しているわけではない。従って、CF 方式ならびに CQ 方式では次のようなデータベースアクセス方法を用いる。

**CF 方式：** 図 1 のように、まず、自サイトの LL 表の情報に従って  $D_i$  が存在すると思われるサイトに処理依頼メッセージを転送する。これを受信したサイトでは、既に  $D_i$  が存在しない場合、送られてきたメッセージを LL 表に従って再転送する。このように、各サイトの LL 表に基づくメッセージ転送が連鎖的に起こった後、最終的に  $D_i$  が存在するサイトで処理依頼メッセージに相当する処理が  $D_i$  に対して行われる。

**CQ 方式：** まず、自サイトの LL 表に従って処理依頼メッセージがあるサイトに転送する。受信したサイトでは、既に  $D_i$  が移動している場合は、そのサイトの LL 表の  $D_i$  の位置情報を問合せ発生サイトへと返送する。問合せ発生サイトでは、返送された情報をもとに自サイトの LL 表の  $D_i$  の位置情報を更新し、更にその新たな位置情報を用いて処理依頼メッセージを再転送する。これを  $D_i$  が存

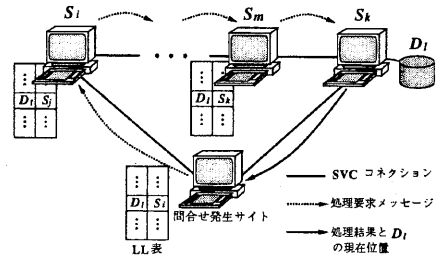


図 1: CF 方式でのメッセージの流れ

在するサイトが見つかるまで繰り返す。

### 2.3 ATM ネットワークの特性を考慮した手法

CF, CQ 方式では、データベースの移動が頻繁な場合に、特定のデータベースへのアクセスにデータベースの移動軌跡に沿った多くのメッセージ通信が必要になり、性能が劣化すると考えられる。

本稿では、この短所を補うことを目的として、新たなデータベース位置管理手法を提案する。提案する手法は、拡張 CF (ECF) 方式と拡張 CQ (ECQ) 方式の二つであり、これらは、ATM ネットワークの特徴である次の二点を利用している。

1. 広い帯域幅により、少量のデータの転送遅延は無視できる。
2. 輻輳が発生した場合を除いて、SVC コネクションは一定時間 (通常 20 分) 不利用にならない限り解放されない。

ECF, ECQ 方式は、データベースアクセスの際の処理依頼メッセージの流れは CF, CQ 方式と同じである。但し、CF, CQ 方式と異なり、LL 表で保持する情報として時刻印が加わる。LL 表の内容は、処理依頼メッセージと一緒に問合せ発生サイトから転送される。LL 表の内容はさほど大きなデータではないため、広帯域ネットワークを用いることによりこの転送遅延は無視できる。ECF, ECQ 方式では次のような LL 表の更新規則を用いる。

更新規則：

- (i) データベース  $D_i$  が、サイト  $S_i$  からサイト  $S_j$  に移動する場合、両サイトの LL 表の  $D_i$  の現在位置を  $S_j$  に書き換え、時刻印をデータベース移動の発生時刻にする。

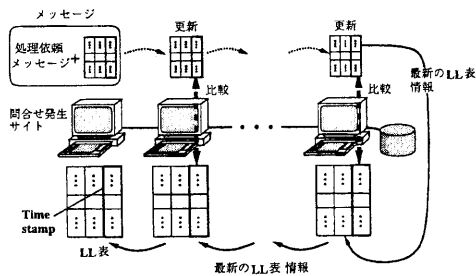


図 2: ECF 方式でのメッセージの流れ

(ii) LL 表の内容を含んだメッセージを受信したサイトは、送られてきた LL 表の内容と自サイトの LL 表の時刻印を各データベースに関して比較する。両者の時刻印が同じでなければ、古い情報の方を新しいものに更新する。

ECF, ECQ 方式では、処理依頼メッセージと一緒に問合せ発生サイトの LL 表の内容も転送される。そして、目的のデータベースが存在するサイトおよびそのサイトまでに経由する各サイトにおいて、上述の (ii) の更新が行なわれる。処理依頼メッセージの転送完了時には、送られてきた LL 表の内容は、問合せ発生サイト、経由したサイト、データベースが存在するサイトのすべてから得られる最も新しい情報となる。

その後、メッセージが転送されてきた SVC コネクションを用いて、逆向きに最新の LL 表情報を、経由した全サイトおよび問合せ発生サイトに転送する。この過程は、新たなコネクション設定を必要としないことから短時間でこなえる。最新の情報を受けとった各サイトは、自サイトの LL 表を更新する。

これにより、各サイトが処理を発生する際に、経由しなければならないサイトの数が減少することが期待される。ECF 方式でのメッセージの流れを図 2 に示す。

### 3 通信所要時間の比較

#### 3.1 比較対象

広帯域ネットワークでは、各手法の性能をトラフィック量よりむしろ、通信所要時間の方が重要と考えられる。本稿では、このような考えに基づ

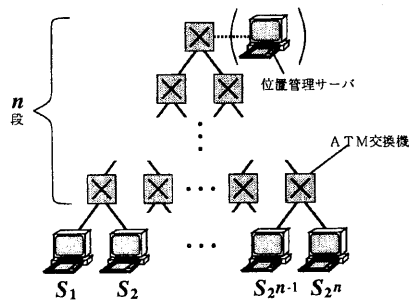


図 3: シミュレーションのネットワークトポロジ

き、各手法の通信所要時間を評価する。

通信所要時間に関しては、データベースアクセスのための通信回数が 1 回ですむことから BN, BF 方式が優位である。しかし、ATM ネットワークにおいてブロードキャストを用いるためには、仮想 LAN を構築しなければならない。これには、マルチキャストサーバ (MCS) と呼ばれるサイトから特定のグループに対して予め一対多通信用の PVC を設定しておく必要がある。世界規模に至るような広域ネットワークでは、分散システム内のサイトの構成が頻繁に変化することから、このような PVC を設定することは非現実的である。

そこで、本稿では、DF, DQ, CF, CQ, ECF, ECQ の六方式を比較の対象とする。

#### 3.2 シミュレーション環境

シミュレーションでは、図 3 のような深さ  $n$ 、サイト数は  $2^n$  ( $S_1, \dots, S_{2^n}$ ) の完全二分木のトポロジをもつ ATM ネットワークを想定する。また、DF, DQ 方式では、位置管理サーバが必要であるため、二分木の根の位置の交換機に位置管理サーバ  $S_0$  を接続する。位置管理サーバ自体はデータベースに対する処理を発生しないものとする。

シミュレーションでは、各手法において、あるサイトが特定のデータベースにアクセスするために処理依頼メッセージを転送し、それに対する返答メッセージを受信するまでの一連の操作の通信所要時間を計算する。但し、データベース移動もデータベース処理の一つと考え、処理がデータベース移動の場合は、DF, DQ 方式において位置管理サーバへの新たな位置情報の通知とそれに対する

Ack メッセージの通知に要する時間を通信所要時間に加える。

### 3.3 シミュレーションの詳細と結果

データベースの総数は 20 ( $D_1, \dots, D_2$ ) とし、それぞれの初期位置をサイト  $S_i$  とする。簡単のために、すべてのサイトは同一確率で問合せを発生し、処理の対象となるデータベースも  $D_1, \dots, D_2$  の中から同一確率で選ばれるものとする。問合せの間隔とデータベース移動の間隔は指数分布に基づき、SVC コネクションは 20 分使用されないと解放されるものとする。

まず、 $n = 5$  とし、平均問合せ間隔を 10 秒から 3 分まで 10 秒ずつ、問合せとデータベース移動の発生比 (問合せ/移動比) を 1 から 15 まで 1 ずつ変化させて、5000 の問合せを各方式で処理した。その結果を図 4 に示す。x 軸が平均問合せ間隔、y 軸が問合せ/移動比、z 軸が一つの問合せに対する平均処理時間を表している。この結果より、DF、DQ 方式は、問合せ/移動比の影響をあまり受けていないのに対し、他の四手法では、問合せ/移動比が小さくなると通信所要時間が大きくなるのが分かる。特に CF、CQ 方式は、問合せ/移動比によって顕著に結果が変化する。

次に、上述の結果から、各問合せ間隔、問合せ/移動比において最適 (通信所要時間が最短) となる手法を求めた。その結果を図 5 (a) に示す。図では、x 軸が平均問合せ間隔、y 軸が問合せ/移動比、図中の各領域が各手法が最適となる範囲を表している。また、 $n = 7$ 、 $n = 10$  の場合にも同様のシミュレーションを行ない、最適手法を調べた。結果をそれぞれ図 5 (b)、(c) にそれぞれ示す。

$n = 5$  では、すべての場合で ECF 方式が最適となる。 $n = 7$  では、問合せ/移動比がおおよそ 5.5 を越える場合、つまりデータベース移動が頻繁には起こらない場合には、ECF 方式が最適となる。それ以外では、問合せ間隔が小さい場合もしくは問合せ/移動比が小さい場合に DQ 方式、その他は DF 方式が最適となる。 $n = 10$  では、ECF 方式が最適となる領域はなくなり、問合せ間隔が小さい場合もしくは問合せ/移動比が小さい場合に

DQ 方式、それ以外では DF 方式が最適となる。

## 4 考察

DF、DQ 方式では、位置管理サーバに正確な位置情報が保持されているため、データベースの移動の影響を殆んど受けない。但し、DF 方式では三つの通信路、DQ 方式では二つの通信路を用いるため、コネクション設定時間の影響は DF 方式が大きい。従って、問合せの頻度が高い場合には、以前に設定した SVC コネクションを利用できる可能性が高いので、DF 方式の性能が良くなる。

一方、その他の四方式では、データベース移動の頻度が高くなると、目的のデータベースが存在するサイトまでに経由しなければならないサイトが増加するため性能が劣化する。また、ネットワークの規模が大きい場合もデータベースの移動が可能なサイトが増加するため性能の低下が著しい。

ECF、ECQ 方式に着目すると、サイトの航行数を削減することによる性能向上の効果が顕著に現れている。これらの手法では、処理に無関係なデータベースの位置情報も更新されるため、システム内のデータベース数が多い場合は、更にその効果が大きくなるものと考えられる。

また、ECF 方式と ECQ 方式を比べると、すべての場合において ECF 方式の方が良い性能を示している。これは、ECQ 方式では、更新された LL 表の情報を経由した各サイトに伝達する際に、必ず問合せ発生サイトを經由する必要があるために、通信回数が増え、性能が低下するからである。

## 5 おわりに

本稿では、データベースの位置管理手法として、従来の手法を応用した手法と ATM ネットワークの特性を考慮した新たな手法を提案した。更に、提案した手法を、通信所要時間の観点からシミュレーションによって比較した。その結果として、問合せとデータベース移動の発生頻度に応じて最適な手法を同定することができた。本稿で得られた結果は、データベースの位置管理だけではなく、広域 ATM ネットワークにおける一般的な資源の位置管理にも応用できる。

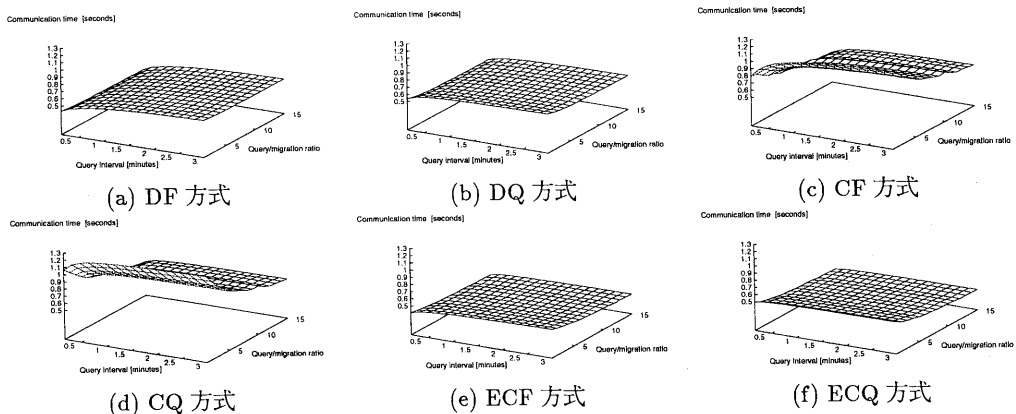


図 4: 各手法の通信所要時間

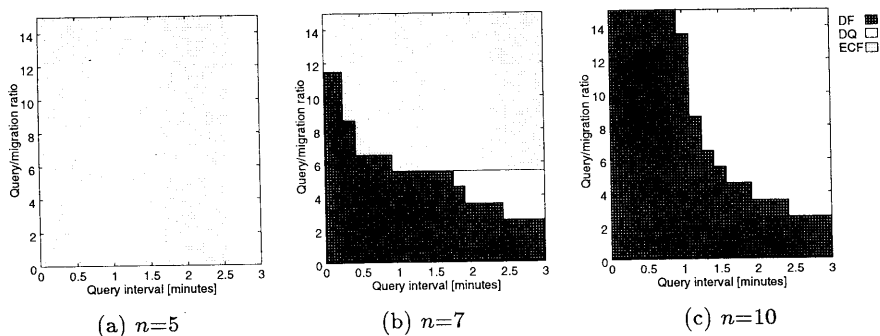


図 5: 最適手法の分布

今後は、より詳しく六手法の比較を行うために、移動先のサイトや問合せを発生するサイトの局在性のある環境や、その局在性が動的に変化するような環境でのシミュレーションを行う予定である。更に、位置管理を考慮した上で、トランザクション処理におけるデータベース移動の有効性を検証していく必要がある。

### 謝辞

本研究の一部は、文部省科学研究費基盤研究(A)(1)(07558040)の研究助成によるものである。ここに記して謝意を表す。

### 参考文献

[1] T. Hara, K. Harumoto, M. Tsukamoto, and S. Nishio, "Database Migration for Transaction Processing in ATM Networks," Department of Information Systems Engineering, Osaka Uni-

versity, Technical Report, ISE-TR-96-004, April 1996.

[2] T. Imielinski and B.R. Badrinath, "Querying in highly mobile distributed environments," Proc. Conf. Very Large Data Bases (VLDB'92), pp.41-52, 1992.

[3] R. Kadobayashi and M. Tsukamoto, "Traffic-based performance comparison of mobile support strategies," ACM-Baltzer Mobile Networks and Nomadic Applications (NOMAD): Topical Journal on Mobility of Systems, Users, Data and Computing, Vol.1, No.1, 1996 (to appear).

[4] 西尾章治郎, 塚本昌彦, "広帯域ネットワークにおけるマルチメディア情報ベース," 信学論 (D-II), Vol.79, No.4, pp.460-467, April 1996.

[5] R. Vingralek, Y. Breitbart, and G. Weikum, "Distributed file organization with scalable cost/performance," Proc. ACM SIGMOD'94, pp.253-264, 1994.