

## ATM 環境におけるデータベース移動に基づく トランザクション処理手法

原 隆浩 春本 要 塚本昌彦 西尾章治郎

大阪大学工学部情報システム工学科

〒565 吹田市山田丘 2-1

{hara,harumoto,tuka,nishio}@ise.eng.osaka-u.ac.jp

近年、ATM 交換方式などの発展により、ネットワークの帯域幅の拡大が進んでいる。このような状況下で、従来はデータ伝送量を削減させることが性能向上の要因であったのに対し、これからはネットワークの帯域幅をいかに有効利用して性能向上を図るかが大きな課題となる。特に、大量のデータの転送を短時間で行なえることから、ネットワークを介したデータベース移動 (database migration) をデータベース処理に利用することも現実的に可能である。そこで、本稿では、ATM ネットワークの仮想 LAN 環境を想定し、従来のトランザクション処理手法とは異なるデータベース移動を用いた処理手法を提案する。また、シミュレーションによって、提案した手法と従来の二相コミットプロトコルを用いたデータベース固定型のトランザクション処理手法との性能比較を行なう。

## Transaction Processing Schemes based on Database Migration on ATM Environments

Takahiro HARA Kaname HARUMOTO Masahiko TSUKAMOTO Shojiro NISHIO

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

2-1 Yamadaoka, Suita, Osaka 565, Japan

{hara,harumoto,tuka,nishio}@ise.eng.osaka-u.ac.jp

Based on the recent development of network technologies such as ATM (Asynchronous Transfer Mode), broader channel bandwidth is becoming available everywhere in the world-wide networks. Considering such status update of network environment, more efficient use of broadband networks is now one of the most important issues for performance improvement of network-wide database systems, though the minimization of transmitted data volume has been the primary factor for the performance improvement in conventional distributed databases. As one of new technologies to make good use of available broadband channel, the database migration through networks becomes practically possible and powerful as one of basic database operations. In this paper, we first propose two new transaction processing methods based on the database migration in virtual LANs (Local Area Networks) with ATM facilities. Then we show simulation results regarding performance comparison between our proposed methods based on the database migration and the conventional method, where the database migration is not assumed and the two-phase commit protocol is employed as one of basic protocols.

## 1 はじめに

近年, ATM (Asynchronous Transfer Mode: 非同期転送モード) 方式を中心として, ネットワーク技術が急速に発展している。特に著しいのが帯域幅の拡大であり, 数百 Mbps や数 Gbps の帯域幅をもつネットワークも普及しつつある。帯域幅の拡大は, 大量のデータの短時間での転送を可能とする。したがって, クライアント・サーバ型のシステムでの応答時間の短縮が期待でき, 特に, 現在盛んに研究が行なわれているサイエンティフィックデータベースや CALS などでは, マルチメディアデータを多く扱うことから, ATM における帯域幅, QoS の保証といった性質が非常に有効となる [4]。移動計算環境においてもバックボーンとして ATM などの広帯域ネットワークを用いることにより柔軟性に優れた高度な通信環境を提供することが可能となる [1, 2, 3, 8]。その他にも, ビデオオンデマンドなど, 多くのマルチメディアアプリケーションが広帯域ネットワーク上で構築されている [5, 9, 10, 13]。

一方, 帯域幅の拡大は, 分散処理において通信コストがボトルネックであったこれまでの状況が一変することを意味している。したがって, 計算機ネットワーク上の分散処理に関する研究は, 通信コストの削減による性能向上から, 帯域幅の有効利用による性能向上を目的とした技術の開発へと移行する必要がある [7]。データベース分野においても, 従来はデータベースは特定のサイトに固定され, それらに対する処理はメッセージによる処理依頼などといったように, 通信コストを削減する手法を開発することが一般的であった。しかし, 広帯域ネットワークでは, 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 (database migration, 以下 DB 移動と呼ぶ) も現実的に可能である。DB 移動を用いることにより, これまでの分散データベース技術とは異なる新たな技術の可能性が出てくる。

例えば, 従来の二相コミットプロトコルによってトランザクション処理を行なった場合, 各サイトに分散するデータベースに対して処理メッセージやその結果の交換を複数回繰り返した後, 二相

コミットのためのメッセージ交換を二往復する必要がある。一方, データベース移動を用いた場合, データベースをトランザクション発生サイトに移動してしまえば, その後のメッセージ交換の必要がない。広帯域ネットワークでは, 大量のデータの転送は短時間で行なえるが, 伝搬遅延は従来のネットワークと比べ大幅に小さいわけではない。したがって, データベース移動によって処理時間の短縮が期待できる。

そこで, 本稿では, 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する。すべての場合においてデータベース移動による処理の方が二相コミットプロトコルを用いたデータ固定型の処理 (以後, 固定処理と呼ぶ) よりも効率的であるとは限らないため, データベースの利用状況やトランザクションの性質に応じて固定処理とデータベース移動の二手法を選択する適応型のプロトコルを考える。また, シミュレーションにより, 提案するプロトコルの性能評価を行なう。以下, 2章で本稿で想定している分散環境を示し, 3章で提案するプロトコルについて説明する。4章では, 性能評価のためのシミュレーションの結果を示し, その結果に対する考察を行なう。最後に5章で本稿のまとめを行なう。

## 2 ATM 環境のモデル

ATM は, 基本的にはコネクション型の交換方式であり, データ交換を行なう前に通信したいサイトとのコネクション (ATM では仮想チャネルという) を確立する必要がある。コネクションの種類としては, ATM 交換機において予めルーティングを設定しておく PVC (Permanent Virtual Channel) と, サイトからの要求によってルーティングを設定する SVC (Switched Virtual Channel) があり, それぞれに一对一通信のためのポイント・ツー・ポイントコネクションと一対多通信のためのポイント・ツー・マルチポイントコネクションがある。PVC を利用する場合は, コネクションの設定時間を省略することができる。

ATM ネットワークでは, あるサイトから複数のサイトへ予めポイント・ツー・マルチポイント

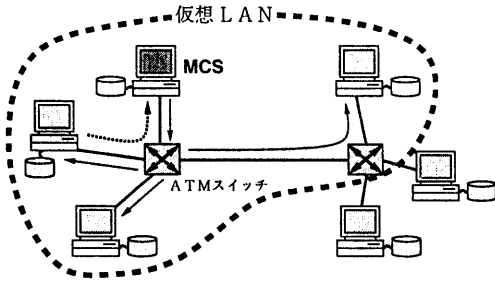


図 1: 仮想LAN とメッセージのブロードキャスト

コネクションのPVCを設定しておくことにより、そのサイトをマルチキャストサーバ(MCS)として特定のグループへのマルチキャストが可能となる。つまり、物理的な接続形態とは異なる仮想的なLANの構築が可能となり、仮想LAN内のデータのブロードキャストは、MCSに対してデータを送信することで実現できる(図1)。これは、基本的にコネクション型のATMネットワークにおいて、仮想LAN内の全サイトへの処理メッセージの送信が容易に行なえることを示している。

本稿では、仮想LAN内の分散データベースに対する分散処理を想定する。したがって、トランザクション発生サイトからのメッセージのブロードキャストには、MCSから仮想LAN内の各サイトに対して設定されているPVCを利用する。データベースが移動する場合、処理に必要なデータベースが存在するサイトと一対一通信を行なおうとすると、各データベースが現在どのサイトに存在するか把握しておく必要がある。そこで、このPVCを用いてメッセージを全サイトにブロードキャストすることにより、データベースの位置管理の手間を省くことができる。その他のメッセージやデータの交換に用いる一対一通信に関しては、必要時にSVCによるコネクションを確立する。ATMでは、輻輳が起こった場合や長い時間(通常20分)その通信路を使用しない場合を除いてコネクションは切断されないため、一度確立されたSVCは一つのトランザクション内で再設定の必要がないものとする。仮想LAN内のデータベースには各々に一意識

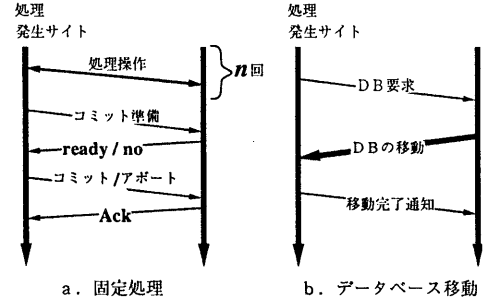


図 2: 各手法の通信手順

別子(DB-id)をもたせる。また、低コストな高速二次記憶アクセスを実現するRAID (Redundant Arrays of Inexpensive Disks) [11] 技術や主記憶データベース技術などにより、高速なデータベースアクセスが可能であると仮定する。

### 3 トランザクション処理プロトコル

本章では、DB移動に基づく適応型トランザクションプロトコルを提案する。また、それらのプロトコルで用いる固定処理とデータベース移動(DB移動)の処理時間の簡単な解析も行なう。

#### 3.1 固定処理, DB移動の処理時間解析

1章で述べたように、従来のトランザクション処理手法を用いると、処理メッセージとその結果の交換( $n$ 回)および、二相コミットのためのメッセージ交換が必要となり、通信回数が多くなる(図2a)。一方、DB移動を用いた場合、通信回数は3回となり、固定処理と比べるとかなり少ない回数となる(図2b)。ただし、想定した環境では、処理を発生するサイトからのメッセージはMCSを介した仮想LAN内のブロードキャストで各サイトに転送される。処理に関わるサイトから処理発生サイトへのメッセージやデータの転送は、SVCによるコネクションを用いて行う。

そこで、後述の適応型トランザクション処理プロトコルに利用するために、固定処理とDB移動の処理時間を定式化する。ただし、この計算がシステムに対して大きな負担とならないように、簡単な式で近似的に表すことを試みる。

### a. 固定処理

固定処理において、処理操作メッセージ ( $n$  回) の交換部分に要する通信時間  $t_m$ 、および、二相コミットメッセージの通信所要時間  $t_s$  は、次のように表される。

$$t_m = n \cdot (d_{MCS}/2 + d_m + M_m/B) + C(k)$$

$d_{MCS}$ : 処理要求サイトから MCS までの平均伝送遅延 (前、後処理を含む)

$d_m$ : 2 サイト間の伝送遅延

$M_m$ : メッセージの平均サイズ

$B$ : ブロードキャスト帯域幅

$k$ : 処理に関わるサイトの総数

$C$ : 処理に関わる全サイトが処理要求サイトとコネクションを設定するのに要する時間 ( $k$  の関数)

$$t_t = t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}$$

$$t_{s_1} = d_{MCS} + d_m + M_{s_1}/B \quad (\text{コミット準備})$$

$$t_{s_2} = d_m + M_{s_2}/B_1 \quad (\text{ready/no})$$

$$t_{s_3} = d_{MCS} + d_m + M_{s_1}/B \quad (\text{コミット/アポート})$$

$$t_{s_4} = d_m + M_{s_2}/B_1 \quad (\text{Ack})$$

$M_{s_i}$ : 各メッセージのサイズ

$B_1$ : 一対一通信の帯域幅

$t_m$  において処理を要求するサイトからのメッセージは、仮想 LAN 内のブロードキャストによって全サイトに送信されるため、MCS (マルチキャストサーバ) までの伝送遅延  $d_{MCS}$  と MCS から各サイトへの伝送遅延  $d_m$  の両方を考慮しなければならない。メッセージ交換  $n$  回中の半数が処理要求側からのメッセージと考えると、 $t_m$  では  $d_{MCS}$  の係数を  $1/2$  とし、 $d_m$  は処理要求と応答の両方に関わるために係数を  $1$  としている。また、MCS から各サイトへの伝送遅延は、処理に関係あるサイトと関係のないサイトがあるためにシステム全体から見た場合の実効値を確定するのは困難である。固定処理における返答や DB の移動といった一対一通信においても、同期的に処理に関係する各サイトから通信が行なわれるのではないため伝送遅延の実効値の決定は困難となる。そこで、MCS への伝送遅延以外は、簡単のためにすべて一定値  $d_m$  と考える。 $C(k)$  は、処理に関係するすべてのサイ

ト (サイト数:  $k$ ) から処理要求サイトへのコネクションを設定するための時間であり、 $k$  の関数としている。しかし、実際は、処理に関係するサイトが処理要求サイトへ初めに返答メッセージを転送する際にコネクションが設定されるため、この値はトランザクションの性質に大きく依存する。

ここで、広帯域ネットワーク (ATM) ということを考慮すると、帯域幅が十分に大きく、各メッセージのサイズは比較的小さいことから  $M_m/B, M_{t_1}/B, M_{t_2}/B_1$  は  $0$  に近似できる。したがって、固定処理の通信所要時間  $T_{fix}$  は次のように表される。

$$\begin{aligned} T_{fix} &= t_m + t_t \\ &= (n+4) \cdot (d_m + \frac{1}{2}d_{MCS}) + C(k) \end{aligned} \quad (1)$$

### b. DB 移動

DB 要求メッセージの転送所要時間を  $t_d$ 、処理に必要なすべてのデータベースの移動所要時間を  $t_{mig}$ 、移動完了メッセージの転送所要時間を  $t_{comp}$  とすると、それぞれ次のように表される。

$$t_d = d_{MCS} + d_m + M_d/B$$

$$t_{mig} = d_m + C(k) + D/B_M$$

$$t_{comp} = d_{MCS} + d_m + M_{comp}/B$$

$M_d$ : DB 要求メッセージサイズ

$D$ : 移動する DB の総サイズ

$B_M$ : DB 移動のための予約帯域幅

$M_{comp}$ : 移動完了通知メッセージサイズ

固定処理同様に  $M_d/B, M_{comp}/B$  を  $0$  と近似すると、DB 移動に要する通信時間  $T_{DB}$  は次のように表される。

$$\begin{aligned} T_{DB} &= t_d + t_{mig} + t_{comp} \\ &= 3d_m + 2d_{MCS} + C(k) + \frac{D}{B_M} \end{aligned} \quad (2)$$

### 3.2 DB 移動に基づくトランザクション処理手法

本節では、3.1 節の解析に基づいて、トランザクション実行時に固定手法と DB 移動で効率的な方を選択する適応型のトランザクション処理プロトコルを提案する。提案するプロトコルは、単純プロトコルと履歴統計プロトコルの二つである。両プロトコルとも、次の 3 ステップで構成される。

### 1. トランザクション解析

処理に必要なデータベース、固定処理でのメッセージ交換回数などを同定する。

### 2. 手法選択

特定の計算により、固定処理もしくはDB移動のどちらを実行するか選択する。

### 3. トランザクション実行

ステップ2で選択された手法を用いてトランザクションを実行する。

以下では、各プロトコルにおける第2ステップの手法選択について説明する。

#### 3.2.1 単純プロトコル

このプロトコルでは、第1ステップ(トランザクション解析)において得られる結果より、3.1節の1,2式の各パラメータを決定して、処理時間の少ない方を選択する。つまり、

$$t_1 = T_{DB} - T_{fix} \quad (3)$$

としたとき、単純プロトコルの第2ステップは次の表される。

```
if  $t_1 < 0$  then select DB 移動
else                select 固定処理
```

ここで、 $T_{DB}$ を計算するためには、移動しなければならないデータベースの総サイズ( $D$ )を知る必要があるため、仮想LAN内の各サイトで、DB-idとそのDB-idをもつデータベースのサイズ情報の表を保持するようにする。これにより、第1ステップで処理に必要なデータベースが同定すれば、 $D$ の値を計算することができる。また、各サイトにおいて、自分のサイトにあるデータベースのサイズが変動する度に全サイトに知らせるのは非効率的であるため、サイズ情報の表で保持している値と実際の値の差が一定値 $\Delta$ を越えた場合のみその値をブロードキャストするようにする。

#### 3.2.2 履歴統計プロトコル

単純プロトコルでは、注目する(一つの)トランザクションを処理する上で二相コミットとDB移動のどちらが効率的かによって手法を選択した。しかし、一つのトランザクションで効率的である選択が、長い目で見た場合は非効率的であるといったことが生じ得る。例えば、(3)式で $t_1 < 0$ となり固定処理が効率的と判断されても、連続して同じサイトからほぼ同様のデータベースを用いるトランザクションが発生する場合には、先にDB移動を選択しておいた方が実際は効率的である。

そこで、ここで提案する履歴統計プロトコルでは、次の三点に注目する。

1. 各データベースは、その利用頻度が(特に最近)高いサイトに位置するべきである。
2. サイトAが特定のデータベースDを連続使用することが分かっている場合、(DのサイトAからの利用頻度が低くても)サイトAがDをもつ優先度を上げるべきである。
3. 分散システム全体として利用頻度の高いデータベースの位置が、システム全体の性能に大きく影響する。

したがって、履歴統計プロトコルでは、単純に式(3)によって手法を選択するのではなく、履歴情報や連続使用の情報を保持することにより、上述のことを考慮した選択を行なう。履歴統計プロトコルで保持するDB情報表を図3に示す。ただし、図中の表の属性は、次の記法を用いている。

**DB-id:** データベース識別子 (DB-id)

**Size:** DB-idで示されるデータベースのサイズ

**Current Loc:** 現在のサイト

**Cnt:** 現サイトが連続使用を指定しているか

0: 連続使用しない, 1: 連続使用する

**Usage-log:** 利用したサイトの履歴

(一列が一つのトランザクションを示す)

DB情報表に必要な情報は、トランザクション毎に発生サイトから全サイトへとブロードキャストされなければならないため、履歴統計プロトコ

DB-id	Size	Current Loc	Cnt	Usage-log		
1	50	A	1	A	...	B   A
2	35	B	0	A   D	...	
⋮	⋮	⋮	⋮	⋮	⋮	
m	40	A	1		...	B   A

図 3: DB 情報表

ルでは、図 2 の各手法の手順の最後に DB 情報通知が追加される。したがって、各手法の処理時間も  $d_{MCS} + d_m$  だけ大きくなる。

データベース  $D$  (DB-id) がサイト  $A$  に存在する有効性を表すものとして、目的関数  $f(A, D)$  を定義する。  $f(A, D)$  は、前述の履歴統計プロトコルが注目している三点のうち第一項で 2、第二項で 1 を反映している。連続使用優先係数  $P$  は、連続使用指定に対しての優先度を表す係数である。

$$f(A, D) = j \cdot P \cdot L + \sum_{i=1}^L \{k_i \cdot (L+1-i)\} \quad (4)$$

$$j = \begin{cases} 1: & A \text{ が } D \text{ の連続使用を指定している あるいは 現トランザクションで } D \text{ を連続利用指定する} \\ 0: & \text{その他} \end{cases}$$

$P$ : 連続使用優先係数  
 $L$ : DB 利用履歴のサイズ (トランザクション  $L$  回分の履歴を残す)

$$k_i = \begin{cases} 1: & i \text{ 回前のトランザクションでサイト } A \text{ が } D \text{ を利用} \\ 0: & \text{その他} \end{cases}$$

次に、注目点の 1, 2, 3 を反映し、データベース  $D$  を現在の位置からサイト  $A$  に移動する有効性を示す評価関数として、移動評価関数  $G(A, D)$  を次のように定義する。ここで、サイト  $A$  のトランザクション  $T_r$  において DB 移動の有効性を示す値  $t_2$  を、 $T_r$  に必要であり、かつ、 $T_r$  の発生サイトに存在しない DB の移動評価関数の平均値として表す。

$$G(A, D) = \frac{L_{use}}{L} \cdot (f(A, D) - f(X, D))$$

$$t_2 = \frac{1}{N_D} \cdot \sum_{D_i \in D_{use}} G(A, D_i) \quad (5)$$

$L_{use}$ :  $L$  回のトランザクションで  $D$  が利用された回数

$X$ :  $D$  の現在位置

$N_D$ :  $T_r$  で利用され  $T_r$  の発生サイトにない DB の個数

$D_{use}$ :  $T_r$  で利用する DB の DB-id の集合

そこで、固定処理と DB 移動のどちらを選択するかを示す値として  $t_{sel}$  を定義すると、履歴統計プロトコルの第 2 ステップは次のようになる。

$$t_{sel} = t_1 - K \cdot t_2 \quad (6)$$

```

if  $t_{sel} < 0$  then select DB 移動
else                select 固定処理

```

$K$  は  $t_2$  の係数であり、ネットワークやシステム管理者などの専門家が、ネットワークやデータベースの規模、アプリケーションの性質に応じて決定する値である。  $K$  の値が、システム全体の性能に大きく影響する。以後、 $K$  を履歴依存係数と呼ぶ。

#### 4 シミュレーション

提案した単純、履歴統計という二つの適応型トランザクション処理プロトコルを、シミュレーションによって性能評価し、固定処理および DB 移動のみを用いた場合と比較する。シミュレーション・プログラムは C++ によって記述する。用いる各パラメータの値および生成条件などを表 1 に示す。

シミュレーションでは、単純プロトコル、履歴統計プロトコル、固定処理のみ、DB 移動のみの各手法で同様のトランザクションを 200 回発生させ、それぞれにおいて手法選択時間と通信時間の和の総和を処理時間として計測していく。シミュレーションは、次の三つの手順からなる。

[パラメータの決定]

まずトランザクション発生頻度に応じてトランザクションの発生サイトを決定する。発生頻度はトランザクション 100 回までは  $p_1$ 、100 回以降は  $p_2$  を使用する。ただし、表 1 の  $p_1, p_2$  の値欄において括弧外の数値は発生頻度、括弧内の数値はその発生頻度をもつサイトのサイト識別子を表して

表 1: パラメータの設定

パラメータ	パラメータの意味	値
$N$	仮想 LAN のサイト数	20 (サイト識別子 1~20)
$N_{DB}$	データベースの総数	20 (DB-id 0~19)
$D_i$	各データベースのサイズ	$40+2i$ [Mbytes] ( $i=0, \dots, 19$ )
$p_1$	トランザクション発生頻度 1	0.025 (1~10), 0.05 (11~18), 0.15 (19), 0.3 (20)
$p_2$	トランザクション発生頻度 2	0.025 (11~20), 0.05 (3~10), 0.15 (2), 0.3 (1)
$Cnt$	トランザクション連続度	0~2
$N_D$	トランザクション $T_r$ で利用するデータベースの数	1~ $N_{DB}$ でランダムに決定 (連続使用指定した DB の数以上)
$D_{use}$	$T_r$ で利用するデータベース	連続使用指定の DB を除いては ランダムに決定 (総数は $N_D$ )
$C_i$	DB 連続使用指定	$D_{use}$ の中からランダムに決定
$d_{MCS}$	MCS への伝搬遅延	0.05 [秒]
$d_m$	その他の 2 サイト間の伝搬遅延	0.1 [秒]
$B_M$	DB 移動のための予約帯域幅	156 [Mbps]
$C$	コネクション設定時間	$C = 0.3k$ [秒] と設定
$n$	固定処理におけるメッセージ交換回数	1~30 でランダムに決定
$L$	利用履歴のサイズ	20
$P$	連続使用優先係数	1
$K$	履歴依存係数	0.5

いる。次に、トランザクションの連続度  $Cnt$ 、トランザクションで利用するデータベースの数  $N_D$ 、利用するデータベース  $D_{use}$  をそれぞれ決定する。 $Cnt$  が 0 以上の場合は、そのサイトが連続的にトランザクションを発生するものとして、以後  $Cnt$  回はそのサイトの発生頻度をもととの発生頻度より 1 ほど高くし、連続使用するデータベースを決定する。これら以外のパラメータ値は、固定値もしくはこれまでに決定したパラメータから自動的に決定するものである。

[手法の選択]…単純、履歴適応プロトコルのみ

パラメータの値に応じてそれぞれ独自の計算を行ない、固定処理もしくは DB 移動を選択する。

[処理実行]

選択した手法による処理に要する通信時間を経過させる。ただし、各手法の通信所要時間は、3.1 節の式 (1),(2) より得られる計算値をそのまま実際の値と考える。

シミュレーションの結果を、図 4 に示す。ただし、グラフにおいて横軸はトランザクション処理数、縦

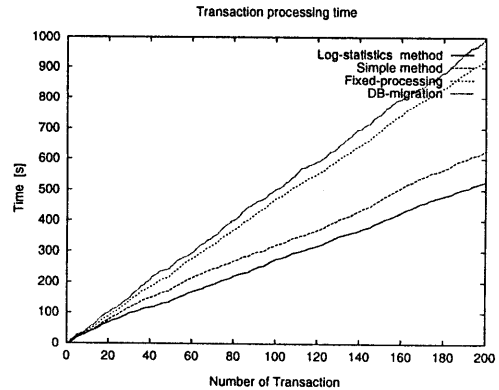


図 4: シミュレーション結果

軸は処理時間の総和である。また、Simple method は単純プロトコルを、Log-statistics method は履歴統計プロトコルを表している。

この結果より、提案した適応型プロトコルの有効性が認められる。特に、履歴適応プロトコルでは、各サイトのデータベースの利用頻度やトランザクションの連続性を考慮して固定処理もしくは

DB 移動を選択するため、単純プロトコルよりよい結果を示している。

データベースの利用頻度が均一な場合やトランザクションの連続性がないような状況では、履歴を考慮することの有効性はなく、メッセージのプロードキャストが一回多い分だけ履歴適応プロトコルの方が単純プロトコルより処理時間がかかり、処理のオーバーヘッドの影響も出てくる。しかし、実際の環境では、利用頻度が均一でなくトランザクションの連続性が存在する方が現実的である。したがって、一般的な環境では履歴適応プロトコルが最も優れていると判断できる。

## 5 おわりに

本稿では、広帯域ネットワーク (特に ATM) の特性を利用してデータベース移動に基づくトランザクション処理手法を提案した。また、シミュレーションにより提案した手法の有効性を検証した。その結果、一般的な環境において履歴適応プロトコルが最も優れているという結果が得られた。現在、提案したプロトコルを実装中である。

今後は、本稿で想定した ATM の仮想 LAN の環境ではなく、メッセージのプロードキャストを利用できない ATM の WAN 環境において同様のトランザクション処理手法を適用することを考えている。この場合に必要となるデータベースの位置管理の手法としては、文献 [6, 12] において提案されている移動体の位置管理手法などが応用できるものと考えられる。また、DB 移動に基づく負荷分散手法についても取り組む予定である。

## 謝辞

本研究の一部は、文部省科学研究費試験研究 (B)(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] Armbrüster, H., The Flexibility of ATM: Supporting Future Multimedia and Mobile Communications, IEEE Communications Magazine, Vol.33, No.2, pp.76-84, Feb. 1995.
- [2] Callender, M.H., Future Public Land Mobile Telecommunication Systems, IEEE Personal Communications, Vol.1, No.4, pp.18-22, 1994.
- [3] Chia, S., The Universal Mobile Telecommunication System, IEEE Communications Magazine, Vol.30, No.12, pp.54-62, Dec. 1992.
- [4] Daily, M., Gabrynowicz, J., Narumalani, S., Nygard, K., Perrizo, W., Ram, P., Reichenbach, S., Seielstad, G.A., and White, W., Accessing Earth System Science Data and Applications Through High-Bandwidth Networks, IEEE Journal on Selected Areas in Communications, Vol.13, No.5, pp.793-805, June 1995.
- [5] Guha, A., Pavan, A., Liu, J., Ajay, R., and Steeves, T., Supporting Real-Time and Multimedia Applications on the Mercuri Testbed, IEEE Journal on Selected Areas in Communications, Vol.13, No.4, pp.749-763, May 1995.
- [6] Kadobayashi, R. and Tsukamoto, M., Performance Comparison of Mobile Support Strategies, Proc. of 1st Int'l Conf. on Mobile Computing and Networking (MobiCom'95), pp.218-225, 1995.
- [7] 西尾章治郎, 塚本昌彦, 春本 要, 下條真司, 広帯域ネットワークにおけるデータベースシステムの可能性, 電子情報通信学会技術研究報告, Vol.95, No.287, pp.49-56, Oct. 1995
- [8] Norp, T. and Roovers, J.M., UMTS Integrated with B-ISDN, IEEE Communications Magazine, Vol.32, No.11, pp.60-65, Nov. 1994.
- [9] Nussbaumer, J.-P., Patel, B.V., Scaffa, F., and Sterbenz, J.P.G., Networking Requirements for Interactive Video on Demand, IEEE Journal on Selected Areas in Communications, Vol.13, No.5, pp.779-787, June 1995.
- [10] Rooholamini, R. and Cherkassky, V., ATM-based Multimedia Servers, IEEE Multimedia, Vol.2, No.1, pp.39-52, Spring 1995.
- [11] Tobagi, F., et al., Streaming RAID - A Disk Array Management System for Video Files, Proc. of ACM Multimedia 93, ACM Press, New York, pp.393-400, 1993.
- [12] Tsukamoto, M., Kadobayashi, R., and Nishio, S., Strategies for Query Processing in Mobile Computing, Imielinski, T. and Korth, H. F. (Eds): Mobile Computing, Kluwer Academic Publishers, pp.595-520, 1996.
- [13] Wetherall, D.J., Stasiar, W.F., Adam, J.F., Houh, H.H., Ismert, M., Bacher, D.R., Phillips, B.M., and Tenninhouse, D.L., View Station Applications: Implications for Network Traffic, IEEE Journal on Selected Areas in Communications, Vol.13, No.5, pp.768-778, June 1995.