

アクセス情報に基づくデータベース移動を用いた データベース再配置手法

秋山 豊和[†] 原 隆浩[†] 春本 要[†]
塚本 昌彦[†] 西尾 章治郎[†]

近年、ネットワークの帯域幅の拡大にともない、分散システムにおいてデータの転送遅延よりもむしろデータの伝播遅延が処理のボトルネックとなってきた。筆者らの研究グループでは、このような分散システムの特性に注目し、データベースの移動（データベース移動）をトランザクション処理に利用することで、データの転送回数を削減し、処理の高速化を実現するトランザクション処理手法を提案している。しかし、これまでに提案した手法はデータベース移動の有効性の検証を目的としていたため、データベースの再配置のために用いるアクセス系列が既知であることを仮定していた。一般にトランザクション開始時にアクセス系列は既知ではないため、実際のシステムに適用するには、トランザクション開始時にサイト間でのアクセスの偏りを検出する機構が必要となる。本論文では、未知のアクセス系列に対して、直前のアクセス情報からサイト間のアクセスの偏りを検出し、データベースの再配置を行う手法を提案する。また、従来のデータベース固定型の処理との性能比較を行い、提案した手法を用いることでデータベース移動を実環境に適用可能であることを示す。

A Database Relocation Method Using Database Migration Based on Access Information

TOYOKAZU AKIYAMA,[†] TAKAHIRO HARA,[†] KANAME HARUMOTO,[†]
MASAHIKO TSUKAMOTO[†] and SHOJIRO NISHIO[†]

Due to the recent development of network technologies, the data propagation delay is becoming a significant factor for the system performance rather than the data transmission delay. Based on this fact, we have proposed a new technology to reduce the number of data transmission by relocating dynamically the databases through networks, which we call *database migration*. Because the proposed method aims to verify the effectiveness of database migration, it was supposed to be able to know the sequence of accesses to the system, which is used for database relocation. However, in order to use database migration in a practical environment, it is necessary to detect the access skew. In this paper, we propose a database relocation method which detects access skew from the access information. Moreover, we examine the effectiveness of the proposed method by comparing with the conventional database-fixed method.

1. はじめに

近年、ATM (Asynchronous Transfer Mode: 非同期転送モード) 方式を中心としたネットワーク技術の発展にともない、ネットワークの帯域幅が急速に拡大している。帯域幅の拡大により広域ネットワークでデータを共有するような分散システムにおいては、ネットワークにデータを送りはじめてからすべてのデータを送り終えるのに要する転送遅延よりも、データが送信元から送信先まで移動するのに要する伝播遅延の方が

処理のボトルネックとなってくる^{1),2)} (図1)。そこで筆者らの研究グループでは、分散データベースにおけるトランザクション処理高速化手法としてデータベース移動を提案している⁷⁾。従来の分散データベースでは、データベースは特定のサイトに固定されており、他のサイトからのアクセスはメッセージを交換することで処理していたが、特定のサイトからのアクセスが集中して発生するような場合には、そのサイトにデータベースを移動してから処理をした方がメッセージ交換の必要がなく高速な処理が可能となる。

さらに、筆者らの研究グループでは、これまでにデータベース移動を用いたトランザクション処理手法を提案している⁵⁾。提案している手法は、トランザクシ

[†] 大阪大学大学院工学研究科情報システム工学専攻
Department of Information Systems Engineering,
Graduate School of Engineering, Osaka University

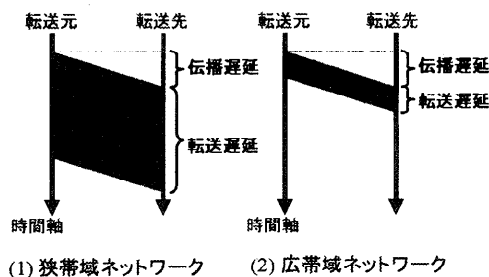


図1 データ転送における遅延時間
Fig.1 Delays of data transmission.

ンに必要なデータベースのサイズ、トランザクションの複雑さ、アクセスパターンなどを考慮して、従来のデータベース固定型の処理（固定処理）とデータベース移動を用いた処理（移動処理）とを適応的に選択する手法である。シミュレーションによって、複数のトランザクションが同時に発生しないような環境における処理効率の評価を行い、提案した手法の有効性を確認している。また、移動中のデータベースに対する処理をサポートすることにより、アクセスの混雑時の並行処理性能の向上を図っている⁶⁾。現在、これらの手法に基づいてデータベース移動を利用した分散データベースシステムの実装を行っている。

これまでに提案している手法は、データベース移動の有効性の検証を目的としていたため、トランザクションのアクセスパターンが既知であると仮定していた。しかし、実環境では一般にトランザクションのアクセスパターンは既知ではないため、これらの手法をそのまま実装中のシステムに適用することはできない。そこで、本論文ではトランザクション開始時に直前のアクセス情報からアクセスの偏りを検出し、固定処理と移動処理を選択する手法を提案する。さらに、提案した手法をシミュレーションによって評価することで、データベース移動の実環境における有効性を検証する。

以下、2章ではこれまでに提案している手法を実環境に適用するうえでの問題点について述べる。3章で本論文において提案する手法について述べる。4章で評価方法および評価結果について述べ、最後に5章で本論文のまとめとする。

2. 従来の手法の問題点

本章では、文献5)で提案したデータベース移動に基づくトランザクション処理手法における手法選択の手順とその問題点について述べる。提案した手法は単純手法と履歴統計手法の2つで、両手法ともトランザクション実行時に固定処理と移動処理のうち効率的な

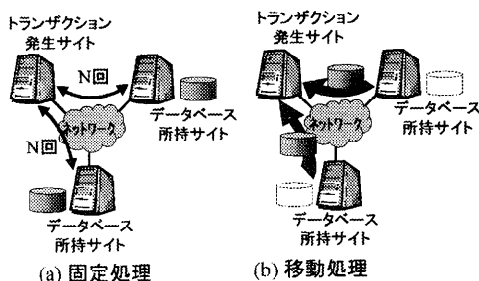


図2 各手法で必要となるサイト間通信
Fig.2 Communication process of the two methods.

方を選択する適応型の処理手法である。

2.1 単純手法

固定処理および移動処理における主なサイト間通信を図2に示す。固定処理では、トランザクション発生サイトからデータベース所持サイトへの処理依頼とデータベース所持サイトからの処理結果の転送が主なサイト間通信である。トランザクションに複数の問合せが含まれる場合、複数回通信が行われる。移動処理では、移動要求、データベースの移動および移動先のサイトからのデータベース移動完了通知が主なサイト間通信である。

単純手法では、移動処理にかかる通信所要時間と固定処理にかかる通信所要時間を見積り、通信所要時間の短い処理手法を選択する。

2.2 履歴統計手法

単純手法では、1つのトランザクションを処理する上で固定処理と移動処理のうち効率的な手法を選択したが、1つのトランザクションで効率的である選択が、長期的には非効率的であるといったことが生じうる。たとえば、実行中のトランザクションだけで判断すると固定処理の方が効率的な場合でも、連続して同じサイトから同一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、先に移動処理をしておいた方が効率的である。

履歴統計手法では、各サイトにデータベースを配置する優先度として次の2つを用いて手法の選択を行う。

- データベースの過去の利用履歴から、特に最近利用頻度が高いサイトに優先的にデータベースを配置する。
- サイト S_i がこの先に特定のデータベース D_j を連続的に使用することが分かっている場合、サイト S_i が D_j を持つ優先度を上げる。

2.3 実環境に適用するうえでの問題点

これらの手法を実環境に適用する場合、次のような問題が生じる。

- 単純手法および履歴統計手法では、トランザクションに含まれる問合せについて、トランザクションの開始時に通信時間を算出する必要があるが、実際のシステムでは受け取った問合せを順に処理していくことになるので、トランザクションの開始時にすべての問合せの通信時間を算出するのは難しい。
- 履歴統計手法ではその時点でアクセスしているサイトがデータベースを継続して使用する場合連続使用を宣言するとしているが、一般に各サイトにおいて次に発生するトランザクションが同じデータベースを利用するか否かは分からない。

このように文献5)の手法には、各サイトが発生するトランザクションのアクセス系列が既知でなければ手法選択のための条件式の算出が困難であるという欠点がある。この手法を実システムに適用するためには、アクセス系列が未知の場合に、直前のアクセス情報を用いて手法を選択する方法を考える必要がある。さらに、これまでの手法では、トランザクションでアクセスするデータベースをすべて移動して処理するか、まったく移動せずに処理するかを選択していたため、場合によっては移動の有効性が低いデータベースも移動してしまう可能性があった。これについても、より柔軟にデータベースごとの選択を行える手法を考える必要がある。

3. アクセス情報を用いた連続系列手法

本章では、アクセス系列が未知の場合に、各時点でのアクセス情報を用いて、各データベースごとに独立して移動を決定する手法(連続系列手法)を提案する。

3.1 データベース移動が有効となるアクセスパターン

本節ではデータベース移動が特に有効となるようなアクセスパターンの例について述べる。データベース移動が有効となるアクセスパターンとしては、以下のようなものが考えられる。

- データベースを集中的にアクセスするサイトが時間とともに順次変移していくような場合。たとえば、グローバルなデータ共有を考えた場合、各サイトでのアクセスパターンは一様であったとしても、サイト間の時差などによって順次、集中的にアクセスするサイトが変わっていくことになる。このような場合、集中的にアクセスするサイトの変化に合わせてデータベースを移動していけば通信コストを大幅に削減できる。
- 各サイトに分割して管理しているデータの統計

処理を特定のサイトで実行するような場合。たとえば、企業における各支社の顧客データおよび売り上げデータについて本社で統計処理を行い、次の戦略を立てる場合が考えられる。統計処理時は多くのデータがやりとりされることになるため、データベースをいったん処理サイトに移動し、処理終了後は元のサイトに戻すようにすれば、通信コストを削減することができる。

このように、データベース移動が特に有効となるのは、特定のサイトでの集中的なアクセスがある程度継続するような場合であることが多い。

3.2 連続系列手法

3.1節で述べたようなアクセスの偏りを検出するための手法として、連続系列手法を提案する。連続系列手法では、各データベースに対して同一のサイトからアクセスする連続したトランザクション列の情報を記録しておき、そのアクセス量がしきい値を超えた場合、その時点でそのサイトに集中的なアクセスが発生していると判断し、データベースを移動する。また、本手法では、文献6)で提案している並行処理制御手法により、移動中のデータベースに対するアクセスはサポートできる。データベースの移動時には、新しい位置情報を各サイトに通知する必要があるが、本論文では移動終了時にデータベースの新しいサイトをブロードキャストで通知する。以下では、各データベースへのアクセス情報の記録およびデータベースの移動をどのように行うかについて述べる。

3.2.1 アクセス情報の記録

本項では、3.1節で述べたようなアクセスの偏りを検出するための情報として、各データベースに対するアクセス情報をどのように記録するかについて述べる。

アクセス情報として以下のような情報をデータベースごとに記録する。

S : 連続したトランザクション列の発生サイト

P_A : 連続したトランザクション列がデータベースに対してアクセスしたページ数

Q : 連続したトランザクション列に含まれる問合せの数

トランザクションの終了時に、終了したトランザクションの情報をそれまでに記録されていたアクセス情報と比較する。終了したトランザクションが記録されているトランザクション列と同一のサイトから発生していた場合、記録されているトランザクション列の P_A 属性および Q 属性に、終了したトランザクションのアクセスしたページ数および問合せ数を加算する。終了したトランザクションがそれまでのトランザクシ

ン列と異なるサイトから発生していた場合には、それまでのアクセス情報は破棄して新たにこのトランザクションの情報をアクセス情報として記録する。

アクセス情報の記録方法としては、特定の1つのサイトで集中的に管理する方法、すべてのサイトで記録する方法、データベースを所持しているサイトに記録する方法などが考えられる。特定のサイトで集中的に管理した場合、アクセス情報の取得時に新たにサイト間の通信が発生してしまう。本論文では、できる限りサイト間の通信回数を削減することを目指しているため、この方法は適切ではない。すべてのサイトで記録した場合、アクセス情報はローカルに取得できる。また、アクセス情報の正確性はそれほど厳密である必要はないので、アクセス情報の更新の通知はトランザクション終了後にトランザクション処理とは独立に行えるため、トランザクションの処理性能にはほとんど影響しない。データベースを所持しているサイトに記録した場合、データベースの移動時にアクセス情報をデータベースとともに移動する必要があるが、アクセス情報のサイズはデータベースサイズと比べると大幅に小さいので、移動時間にはほとんど影響しない。アクセス情報の取得時にはデータベース所持サイトと通信する必要があるが、データベースの施錠要求や、問合せの解析に必要なデータベースの属性情報などのメタ情報の要求の応答メッセージにアクセス情報を付加して転送することができる。また、アクセス情報の更新はローカルに実行できるため、アクセス情報を処理するための付加的なコストはきわめて小さい。

以上のように、アクセス情報は、すべてのサイトで記録するか、データベースを所持しているサイトに記録することで、トランザクションの処理性能に影響しない程度のコストで管理することができる。

3.2.2 データベースの移動

本項では、3.2.1項で述べたアクセス情報に基づいて、どのように各データベースの移動を決定するかを述べる。移動を決定するタイミングは従来の手法と同様にトランザクションの開始時とする。従来の手法では、トランザクションごとにすべてのデータベースを移動して処理するか、固定して処理するかを選択していたが、本手法では各トランザクションにおいて、データベースごとに移動するか否かを選択する。これによって、アクセスパターンが異なるデータベースを同一のトランザクションで利用するような場合、移動する有効性の高いデータベースだけを移動できる。

各データベースの移動は以下のような手順で実行する。

- (1) 到着したトランザクションがアクセスするデータベースの中で、ローカルサイトにないデータベースについて移動するか否かを選択する。
- (2) ローカルサイトにないデータベースのアクセス情報に記述されている、連続して同一サイトから発生した直前のトランザクション列が、次に処理するトランザクションと同じサイトから発生していた場合、そのトランザクション列に対してデータベースを移動せずに処理した場合のデータベース操作にかかる通信時間 (T_{fix}) およびデータベースを移動して処理した場合の通信時間 (T_{DB}) を求める。

ここで、 T_{fix} および T_{DB} は次のように求める。

$$T_{fix} = P_A \cdot D_T + 2Q \cdot D_P$$

$$T_{DB} = P_{DB} \cdot D_T + 3D_P$$

P_{DB} : 注目しているデータベースのページ数

D_T : データをネットワークに送り出すときに生じる遅延 (転送遅延)

D_P : データがネットワーク内を伝わる遅延 (伝播遅延)

T_{fix} の第2項は固定処理では問合せおよびその結果の転送回数だけ伝播遅延が生じることを示している。 T_{DB} の第2項は移動要求、データベースの移動、および移動完了通知による伝播遅延を示している。 T_{fix} では第2項が、 T_{DB} では第1項が支配的となる。

- (3) T_{fix} が T_{DB} よりも大きくなる場合、注目しているデータベースをトランザクション発生サイトに移動する。これは各時点において、同一サイトで連続して発生しているトランザクション列のうち、すでに処理したものについて、データベース移動を用いて処理した方が通信時間が短くなる場合、アクセスに偏りが生じており、さらに連続したアクセスが発生するものと判断してデータベースの移動を決定することを示している。

4. シミュレーションによる性能評価

本章ではシミュレーションによって、従来のデータベース固定型の処理と本論文で提案した手法による適応型の処理の性能比較を行い、提案した手法の有効性を検証する。シミュレーションでは、待ち行列モデルを用いて評価を行った。分散データベースシステムの待ち行列モデルによる評価の有効性は、文献8), 9)などに示されている。

トランザクションの平均応答時間を比較することで評価を行う。文献6)で提案している並行処理制御手法によって、データベースの移動時にもトランザクションを処理できるが、移動中に更新を許可する場合、どちらのサイトに更新を許可するか、更新内容の通知、更新許可サイトの変更をどのタイミングで行うのかという点を考慮する必要があるため、本論文では簡単のため移動時には読み出しのみ許可することにする。また、3.2.1項で述べたように、アクセス情報の処理コストは他の処理コストに比べて十分に小さいため、本シミュレーションでは考慮しない。データベースの移動通知は3.2節で述べたようにブロードキャストで行うが、本シミュレーションで想定する環境ではサイトの構成が対称的であるため、ブロードキャストに要する時間をサイト間の通信に要する時間と同じ値に設定した。

シミュレーションでは、次の3つの手法を比較する。**MIG**：すべてのトランザクションを移動処理のみで処理する手法。トランザクション開始時にローカルサイトにないデータベースをすべてローカルサイトに移動してから処理する。

FIX：すべてのトランザクションを固定処理のみで処理する従来の手法。

SEL：本論文で提案した手法。各データベースを移動するか否かを適応的に選択してトランザクションを処理する。

ここで、文献5)で提案した手法はアクセスパターンが既知であることを仮定しており、本論文で提案した手法とは前提条件が異なるため、本シミュレーションでは比較対象としない。

4.1 シミュレーション環境

シミュレーションでは、3.1節で述べたようなアクセスパターンでデータベースが利用される場合に、3つの手法の平均応答時間を比較する。シミュレーション環境を図3に示す。アクセスするデータベースがトランザクション発生サイトにない場合、トランザクションのうち、サーバで解析された問合せの部分問合せ、すなわち実際にデータベースにアクセスする部分のみの処理をデータベースを所持するサイトに依頼し、残りの部分はクライアントからトランザクションを受け取ったサーバで処理するものと仮定する。トランザクションは、移動処理ですべてのデータベースがトランザクション発生サイトに移動した場合にはトランザクション発生サイトで処理され、そうでない場合はトランザクション発生サイトおよびデータベース所持サイトで処理される。シミュレーションで用いるパラメー

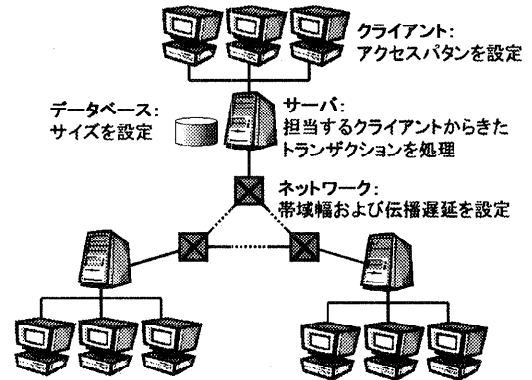


図3 シミュレーション環境

Fig. 3 Simulation environment.

表1 シミュレーションのパラメータ

Table 1 Parameters for simulation.

パラメータ	値
サイト数	3
DB数	1
DBサイズ	611 ~ 18311 ページ (5 ~ 150 Mbytes)
帯域幅 (=転送遅延 (D_T))	1 G ~ 10 Mbps (66 ~ 6554 μ sec/ページ)
伝播遅延 (D_P)	10 ~ 400 msec
メモリアクセス速度	10 nsec/ページ
ディスクアクセス速度	30 msec/ページ

タの値を表1に示す。本シミュレーションでは、データベースとして主記憶データベース^{3),4)}を想定する。I/Oに関しては、読み出し操作の場合はメモリアクセスのみを、書き込み操作の場合はメモリアクセスと更新内容のログを記録するためのディスクアクセスを考慮する。サイト数は3に固定したが、提案した手法はサイト数によらないので、アクセスパターンの特徴が変わらなければサイト数を変化させても同様の結果が得られるものと考えられる。また、簡単のためにデータベース数は1としたが、これについても手法の選択はデータベースごとに行うので、データベース数が増加しても同様の結果が得られるものと考えられる。I/Oの単位をページとしたので、データベースのサイズはページ単位で表記している。1ページは8192バイトとしている。

トランザクションのアクセスパターンは、表2に示したようなパラメータで表す。各サイトに属するクライアントから発生するトランザクションの到着間隔は指数分布に従うものとし、その平均到着間隔をパラメータとして与える。集中的なアクセスを発生させるために、アクセスパターンに周期性を持たせ、トランザクションの平均到着間隔を周期ごとに変化させる。

表2 アクセスパターンのパラメータ

Table 2 Parameters of access pattern for simulation.

パラメータ	値
平均到着間隔	5 ~ 120 秒
到着間隔変更周期	20 ~ 1600 秒
到着間隔比	2 ~ ∞
問合せ/トランザクション	1 ~ 20
問合せのアクセスするデータ量	1/40 ~ 1

このとき集中的にトランザクションが発生するサイトは一様分布の乱数によって決定するので、各サイトの総アクセス量は均一となる。そのため、FIXで処理する場合、データベースはどこに配置しても同様の結果となる。本論文ではデータベースの初期配置は3手法とも同じサイトで評価した。さらに、手法選択に影響すると考えられる各サイトでのトランザクションの到着間隔の比、トランザクションに含まれる問合せ数、各問合せがアクセスするデータ量をパラメータとして与える。トランザクションの到着間隔の比は、通常のトランザクションの到着間隔が、集中的に発生するときの到着間隔の何倍になっているかで表す。問合せのアクセスするデータ量は、データベースの総ページ数に対する割合で表し、問合せごとにページ数1から指定したページ数の間でランダムに決定している。書き込み操作は、3トランザクションに1回発生するようにしている。書き込み操作の問合せのアクセスするデータ量は、データベースの総ページ数の1/50に固定している。

4.2 実験1：帯域幅

帯域幅を変化させたときの各手法の平均応答時間を図4に示す。帯域幅以外のパラメータは表3、表4のように設定した。図4において、MIGの帯域幅10Mbpsの結果は、58秒程度と非常に大きい値になったので省略した。図から分かるように、100Mbpsより帯域幅が広い場合には提案した手法SELが良い結果を示している。また、帯域幅が狭い部分ではデータベース固定型の手法FIXの方が良くなっている。帯域幅が狭いと、少々トランザクションが連続して発生してもデータベースの移動に時間がかかるため移動処理が選択されることはない。そのため、SELとFIXはほぼ同じ平均応答時間になるはずであるが、ごく稀に発生する長期にわたって連続したトランザクションによって移動処理が選択され、そのときの移動のオーバーヘッドによって平均応答時間が悪くなるものと考えられる。帯域幅が600Mbpsを超えると移動処理のみの手法MIGがFIXよりも良くなっている。

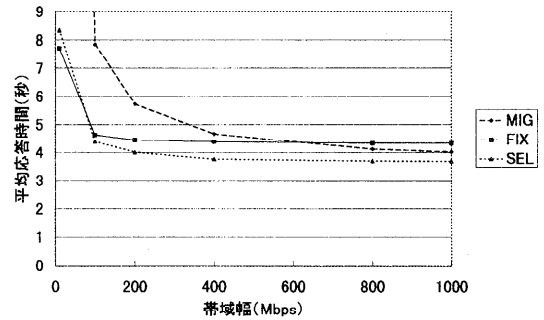


図4 帯域幅と平均応答時間

Fig. 4 Bandwidth and average response time.

表3 実験1のパラメータ

Table 3 Parameters for the experiment 1.

パラメータ	値
サイト数	3
DB数	1
DBサイズ	12220 ページ (100 Mbytes)
伝播遅延 (D_P)	200 msec
メモリアクセス速度	10 nsec/ページ
ディスクアクセス速度	30 msec/ページ

表4 実験1のアクセスパターンのパラメータ

Table 4 Parameters of access pattern for the experiment 1.

パラメータ	値
平均到着間隔	(集中時) 20 秒 (散発時) 120 秒
到着間隔変更周期	400 秒
到着間隔比	6
問合せ/トランザクション	3
問合せのアクセスするデータ量	1/20

4.3 実験2：データベースのサイズ

データベースのサイズを変化させたときの各手法の平均応答時間を図5に示す。帯域幅、データベースのサイズ以外のパラメータは実験1のものと同じで、帯域幅は400Mbps (164 μ sec/ページ)とした。以降の実験においても、変化させるパラメータ以外は実験1と同じ値を用い、帯域幅は400Mbpsとする。データベースのサイズが非常に小さくなると、データベースの移動にかかる時間が短くなり、問合せの依頼および結果の転送にかかる時間と変わらなくなるため、アクセスが集中したときのみ移動するSELよりも、つねに移動して処理するMIGの方が通信回数が少なくなり、良い結果を示している。しかし、実際にこのようなサイズのデータベースを利用することは少ないので、ほぼどのサイズでもSELが良い結果を示しているといえる。データベースのサイズが大きくなると、

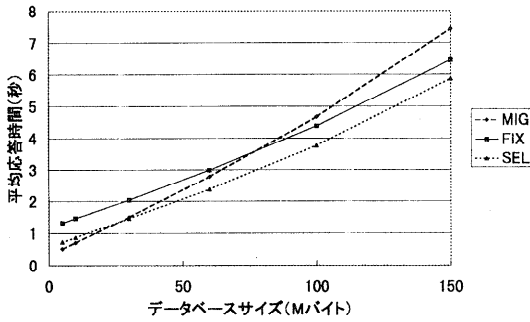


図5 DBサイズと平均応答時間

Fig. 5 Database size and average response time.

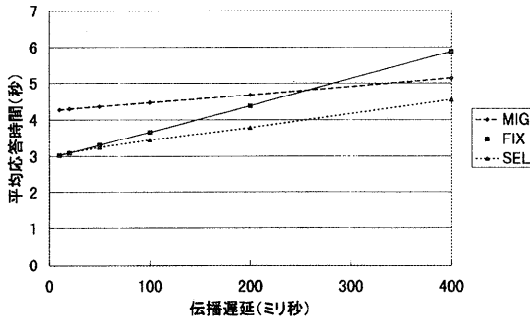


図6 伝播遅延と平均応答時間

Fig. 6 Propagation delay and average response time.

MIGがFIXよりも悪くなっている。このようなことから、本論文で提案しているようなアクセス情報に応じてデータベースを移動するかしないかを選択する手法が有効であると考えられる。グラフを見やすくするために省略したが、さらに大きなサイズのデータベースを用いた場合、およそ300MバイトでFIXとSELが同程度の平均応答時間となった。データベースサイズが大きくなるとSELは移動処理を行わなくなるので、FIXより悪くなることはない。ただし、3.2.2項で述べた T_{fix} および T_{DB} の値がほぼ等しくなるような、移動の決定が微妙な部分では、移動処理を選択して平均応答時間が悪くなるのが稀にある。

4.4 実験3：伝播遅延

伝播遅延を変化させたときの各手法の平均応答時間を図6に示す。伝播遅延が10~20ミリ秒程度の場合、FIXとSELはほぼ同じ平均応答時間を示しており、データベース移動を行っていないことが分かる。伝播遅延が250ミリ秒付近でMIGとFIXの処理時間が逆転している。このグラフから、データベース移動が有効となるのは、伝播遅延がおよそ200ミリ秒以上の場合であることが分かる。世界規模のネットワークでも400ミリ秒より大きな伝播遅延を生じることはほと

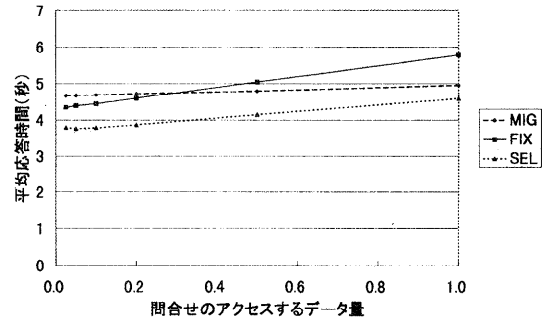


図7 問合せのアクセスするデータ量と平均応答時間

Fig. 7 Amount of data accessed by queries and average response time.

んどないので、200~400ミリ秒で有効となる。

4.5 実験4：問合せのアクセスするデータ量

問合せのアクセスするデータ量を変化させたときの各手法の平均応答時間を図7に示す。アクセスするデータ量が増加するほど、固定型の手法FIXに対してデータベース移動を用いたMIGおよびSEL手法が良くなっている。アクセスするデータ量がデータベースのサイズに近づくと、1つのトランザクションに含まれる問合せ数が単一ではないため、トランザクションに含まれる問合せの結果のデータ転送量よりもデータベース移動のためのデータ転送量の方が少なくなり、結果としてデータベース移動を用いた方が平均応答時間を短くできる。このようにデータベース移動を用いれば、トランザクション処理時間だけでなく場合によってはデータ転送量を削減することも可能となる。アクセスするデータ量がデータベースのサイズに近くなる場合としては、統計処理を行う場合や、インデックスの作成されていない属性に関してアクセスする場合は考えられる。SEL手法の平均応答時間は、アクセスするデータ量に比例して増加しているが、これは各サイトにおける問合せの処理時間の増加によるものであり、アクセスするデータ量の手法選択に与える影響はそれほど大きくないものと考えられる。そのため、SELがMIGおよびFIXに対してつねに良い結果を示している。

4.6 実験5：トランザクションに含まれる問合せ数

トランザクションに含まれる問合せ数を変化させたときの各手法の平均応答時間を図8に示す。トランザクションに含まれる問合せ数が少ない場合、トランザクションが連続して発生してもサイト間の通信にはそれほど時間がかからないので、SELにおいて移動はほとんど発生せず、FIXとSELの平均応答時間がほぼ等しくなっている。トランザクションに含まれる問合せ

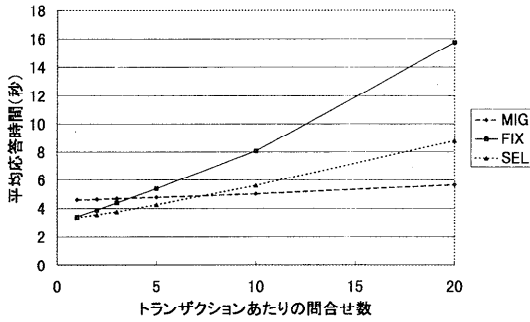


図 8 問合せ数と平均応答時間

Fig. 8 Number of queries and average response time.

せ数が増加すると、トランザクションがそれほど連続して発生しなくてもデータベース移動が有効となるようなアクセスが発生するので、SEL および MIG が良い結果を示している。

トランザクションに含まれる問合せ数が 7 を超えたあたりで SEL と MIG の平均応答時間が逆転している。SEL ではあるサイトからの連続したアクセスによって生じる通信所要時間が、同じ処理にデータベース移動を用いた場合に生じる通信所要時間を上回るような場合、そのサイトでさらにアクセスが連続して発生するという仮定のもとに移動を行っている。ところが、トランザクションに含まれる問合せ数の増加により、単一のトランザクションによるアクセスがこの条件を満たすような場合、散発的に発生したトランザクションであったとしても、さらにトランザクションが継続して発生するものとして、次にそのサイトからトランザクションが発生するとデータベース移動を用いてトランザクションを処理してしまう。これによってあまり効果的でない移動が発生し、SEL の性能を低下させている。また、トランザクションに含まれる問合せ数が増加すると、すべてのトランザクションをデータベース移動を用いて処理した方が良くなるので、アクセスの偏りを検出する期間が必要な SEL よりも MIG の方が良い結果を示している。

以上のように、トランザクションに含まれる問合せ数が非常に多くなると、提案した手法 SEL の性能は低下してしまう。ここで、このように非常に多くの問合せを含むトランザクションは、特定のアプリケーションの操作によって生成されることが多いと考えられる。そこで、アプリケーションの操作ごとに記録した統計情報を利用することで、このようなトランザクションを検出し、より効率的なデータベース移動を実現することができる。この詳細については 4.9 節で述べる。

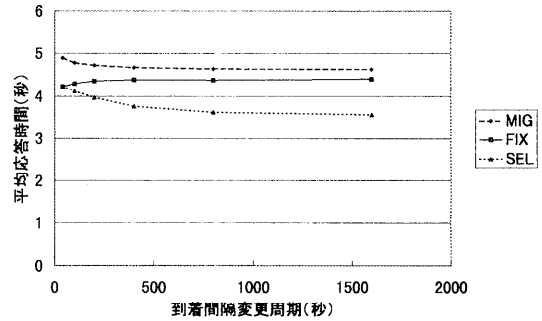


図 9 到着間隔変更周期と平均応答時間

Fig. 9 Period to change the arriving interval and average response time.

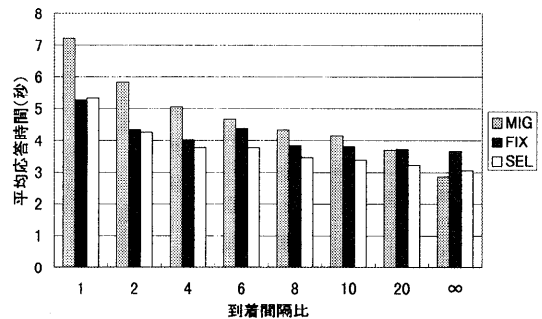


図 10 到着間隔比と平均応答時間

Fig. 10 Ratio of arriving interval and average response time.

4.7 実験 6：到着間隔変更周期

トランザクションの到着間隔変更周期を変化させたときの各手法の平均応答時間を図 9 に示す。到着間隔変更周期が短く、集中発生時の平均到着間隔とはほぼ同じ程度の場合、集中的に発生するトランザクションはほとんどないので、データベース移動は行われず、SEL と FIX はほぼ同じ平均応答時間になっている。また SEL はアクセスの偏りの検出時にある程度の期間を必要とするので、到着間隔変更周期がより長いほどアクセスの偏り検出のためのオーバーヘッドが少なくなり、良い結果を示している。

4.8 実験 7：到着間隔比

トランザクションの到着間隔比を変化させたときの各手法の平均応答時間を図 10 に示す。基準となる集中的に発生しているときのトランザクションの平均到着間隔を 20 秒に固定したので、総トランザクション数および単位時間あたりのトランザクション数は到着間隔比が大きくなるほど少なくなっている。これにより、全体として右下がりのグラフとなっている。到着間隔比 ∞ は、あるサイトで集中的にトランザクシ

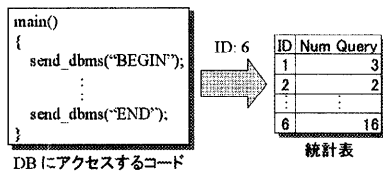


図 11 アプリケーションから DB へのアクセス
Fig. 11 Access from application to database.

ンが発生している間、他のサイトではトランザクションは発生しないことを示す。集中的なアクセスが発生しない場合（到着間隔比 = 1）やアクセス集中時に他のサイトから散発的なアクセスが発生しない場合（到着間隔比 = ∞ ）などの特殊な場合を除けば、提案した手法 SEL が良い結果を示している。

4.9 考 察

提案した手法を用いれば、多くの環境下において、少なくとも従来のデータベース固定型の手法およびデータベース移動のみを用いた手法よりも短い平均応答時間でトランザクションを処理できることが分かった。しかし、4.6 節で示したようにトランザクションに含まれる問合せ数が非常に多くなるとデータベース移動のみを用いた手法の方が良い結果を示す。このような状況に対処するために、トランザクションに含まれる問合せ数を予測する方法として、以下のようなものが考えられる。

アプリケーションからのデータベースへのアクセスは、図 11 に示すようにデータベースアクセス用の言語がアプリケーションのコードに埋め込まれていることが多い。そのため、あるアプリケーションの操作によって生じるデータベースアクセスは同様なものになることが多い。このような特性を利用することで、次に処理するトランザクションの特徴を予測することができる。すなわち、これらのアプリケーションの操作に ID を割り当て、操作ごとにトランザクションの特徴を記録し、次に処理するトランザクションを生成した操作について過去のアクセスの特徴を調べ、その特徴をもとにしてデータベースの移動を決定する。このようにすることで、個々のトランザクションに含まれる問合せ数が多い場合にも対応することができる。この手法を評価するためには、実環境に基づいたアクセスパターンのモデルが必要となるため、今後このようなモデルについても考える必要がある。

さらに、本論文では比較的単純な手法で次のアクセスを予測していたが、過去のアクセス系列情報から文献 10)~12) など提案されている手法で、アクセスパターンを抽出し、それをもとに次のアクセス系列を

予測する手法なども考えられる。このような手法を用いることで、各サイトで発生する操作間に相関性があるような場合、集中的なアクセスが発生する前にデータベースを移動できるため、通信時間の削減が期待できる。しかし、この手法では過去の系列からのアクセスパターンの抽出および各時点で生成中のアクセスパターンの識別など実現が困難な点がある。これらの点については今後適切な手法を考えていく必要がある。

5. おわりに

本論文では、データベース移動を用いた分散データベースシステムにおいて、トランザクションのアクセス情報を用いて適応的にデータベースの移動を決定する手法を提案した。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。その結果より、提案した手法を用いればアクセス系列が既知でなくともデータベース移動を有効に利用できることが分かった。今後は、実装中のシステムに提案した手法を適用し、プロトコルオーバーヘッドなどを含めた実測評価を行う必要がある。

謝辞 本研究は、文部省科学研究費奨励研究 (A) (09780380), (A) (10780260) および日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(Project No.JSPS-RFTF97P00501) 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) Banerjee, S., Li, V.O.K. and Wang, C.: Distributed Database Systems in High-Speed Wide-Area Networks, *IEEE Journal on Selected Areas in Commun.*, Vol.11, No.4, pp.617-630 (1993).
- 2) Banerjee, S. and Panos, C.K.: Network Latency Optimizations in Distributed Database Systems, *Proc. 14th Int'l Conf. on Data Engineering*, pp.532-540 (Feb. 1998).
- 3) DeWitt, D., Katz, R., Olken, F., Shapiro, L., Stonebraker, M. and Wood, D.: Implementation Techniques for Main Memory Database Systems, *Proc. ACM SIGMOD '84*, pp.1-8 (June 1984).
- 4) Garcia-Molina, H., Lipton, R.J. and Valdes, J.: A Massive Memory Machine, *IEEE Trans. Comput.*, Vol.C-33, pp.391-399 (1984).
- 5) 原 隆浩, 春本 要, 塚本昌彦, 西尾章治郎: データベース移動を用いた ATM ネットワークにおけるトランザクション処理, 信学論 (D-I), Vol.J80-D-I, No.6, pp.505-513 (1997).

- 6) 原 隆浩, 春本 要, 塚本昌彦, 西尾章治郎: データベース移動に基づく分散データベースシステムとその並行処理制御機構, 信学論 (D-I), Vol.J81-D-I, No.6, pp.819-828 (1998).
- 7) 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 (1996).
- 8) Sheikh, F. and Woodside, M.: Layered Analytic Performance Modelling of a Distributed Database System, *Proc. 17th Int'l Conf. on Distributed Computing Systems*, pp.482-490 (May 1997).
- 9) Jenq, B., Kohler, W.H. and Towsley, D.: A Queueing Network Model for a Distributed Database Testbed System, *IEEE Trans. Softw. Eng.*, Vol.14, No.7, pp.908-921 (1988).
- 10) Dietterich, T.G. and Michaelski, R.S.: Discovering Patterns in Sequences of Events, *Artificial Intelligence*, Vol.25, pp.187-232 (1985).
- 11) Kudenko, D. and Hirs, H.: Feature Generation for Sequence Categorization, *Proc. 15th National Conf. on Artificial Intelligence, 10th Conf. on Innovative Applications of Artificial Intelligence*, pp.733-738 (1998).
- 12) An, A., Chan, C., Shan, N., Cercone, N. and Ziarko, W.: Applying Knowledge Discovery to Predict Water-Supply Consumption, *IEEE Intelligent Systems*, Vol.12, No.4 (1997).

付 録

A.1 シミュレーションで用いた待ち行列モデル
各サイトのシステムを, 図 12 に示されるような待ち行列モデルでシミュレートした. 図中の記法としては, 次のものを用いた.

- CLI:** クライアント
TM: トランザクションマネージャ
RL: 読み出し用施錠
WL: 書き込み用施錠
NET: ネットワーク
MEM: 主記憶
DISK: ディスク

システムは以下のように動作する. クライアントから発生したトランザクションは, まずトランザクションマネージャに渡される. トランザクションに含まれる問合せが同じサイトにあるデータベースにアクセスする場合は, ローカルなデータベースに施錠をする. そうでない場合は, ネットワークを通して他のサイトに存在するデータベースに施錠する. 読み出し施錠を確保した場合, 主記憶上のデータにアクセスし, 終

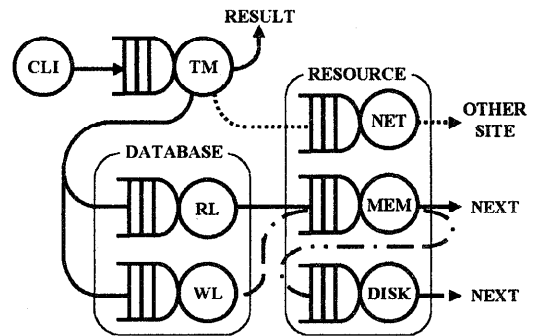


図 12 待ち行列モデル

Fig. 12 Queuing model of each site.

了したらトランザクションマネージャは次の問合せの処理を開始する. 書き込み施錠を確保した場合, 主記憶上のデータにアクセスした後, ディスク上にログを記述し, 終了したら次の問合せの処理に移る. 本論文のシミュレーションでは, 問合せの解析や最適化について考慮していないため, 待ち行列モデルにおいても CPU での処理時間は考慮していない.

(平成 10 年 8 月 17 日受付)

(平成 11 年 3 月 5 日採録)



秋山 豊和 (学生会員)

1997 年大阪大学工学部情報システム工学科卒業. 同大学院工学研究科博士前期課程所属. データベースシステム, 分散処理に興味を持つ. 電子情報通信学会会員.



原 隆浩 (正会員)

1995 年大阪大学工学部情報システム工学科卒業. 1997 年同大学院工学研究科博士前期課程修了. 同年, 同大学院工学研究科博士後期課程中退後, 同大学院工学研究科情報システム工学専攻助手となり, 現在に至る. 1996 年本学会山下記念研究賞受賞. データベースシステム, 分散処理に興味を持つ. IEEE, 電子情報通信学会各会員.

**春本 要 (正会員)**

1992年大阪大学基礎工学部情報工学科卒業。1994年同大学院基礎工学研究科博士前期課程修了。同年、大阪大学大学院工学研究科情報システム工学専攻助手となり現在に至る。

データベースシステム、マルチメディア情報システムの研究に従事。IEEE, 電子情報通信学会各会員。

**塚本 昌彦 (正会員)**

1987年京都大学工学部数理工学科卒業。1989年同大学院工学研究科修士課程修了。同年、シャープ(株)入社。1995年大阪大学大学院工学研究科情報システム工学専攻講師。1996

年より、同大学院工学研究科情報システム工学専攻助教授、現在に至る。工学博士。時空間データベースおよびモバイルコンピューティングに興味を持つ。ACM, IEEE等7学会の会員。

**西尾章治郎 (正会員)**

1975年京都大学工学部数理工学科卒業。1980年同大学院工学研究科博士課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教授を経て、1992年より大阪大学大学院工学研究科情報システム工学専攻教授となり、現在に至る。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データ

ベース、知識ベース、分散システムの研究に従事。現在、Data & Knowledge Engineering, Data Mining and Knowledge Discovery, The VLDB Journal等の論文誌編集委員。ACM, IEEE等8学会の会員。