研究会推薦論文

データベース移動に基づく分散データベースシステム DB–MAN α の設計と実装

酒井 仁[†] 海貝明道[††] 秋山豊和[†††]
原 隆浩[††] 塚本昌彦[††] 西尾章治郎[††]

近年，ネットワークの帯域幅が拡大し，分散システムでは，データの転送遅延よりデータの伝播遅延が処理時間に大きな影響を与えるようになってきている．筆者らの研究グループでは，分散データベースシステムにおいて通信回数を削減し，伝播遅延の影響を小さくするため，データベースを移動してトランザクションを処理する手法（データベース移動）を提案している．さらに，トランザクションの複雑さなどによっては従来のデータベース固定型の処理の方が処理時間を短くできる場合があることを考慮して，トランザクション処理方法を適応的に選択する手法も提案している．本論文では，これらの提案に基づいて，移動機能を有する分散データベースシステムを設計し，そのプロトタイプシステム DB–MANα を実装する．DB–MANα は，主記憶データベースを用いてデータベース移動を高速化するため，データベースの移動を考慮したバックアップ管理機構を有している．本論文では，実装したシステムの実測評価を行い，システムの実環境における有効性を示す．

DB–MANα: A Distributed Database System Based on Database Migration

Shinobu Sakai,[†] Akimichi Umigai,[††] Toyokazu Akiyama,[†††]
Takahiro Hara,[††] Masahiko Tsukamoto[††] and Shojiro Nishio[††]

Due to the recent expansion of network bandwidth, the data propagation delay has become a significant factor which influences the system performance in place of the data transmission delay. Based on this fact, we have proposed a new technology to reduce the bad influence of propagation delay on the system performance by relocating dynamically the database through networks, which we call *database migration*. Furthermore, we have proposed a database relocation method to choose the transaction processing method between the conventional database fixed method and the proposed database migration method by giving consideration to the transaction complexity. In this paper, we explain our distributed database system with database migration mechanism based on these proposals, and the implementation of the DB–MANα system as a prototype system. The DB–MANα system reduces the database migration time by using a main memory database technique, which induces us to add a backup management mechanism for migratory databases. We show some measurement results for the performance evaluation of the DB–MANα system.

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，それにともなってデータ通信に用いることのできるネットワークの帯域幅も急速に拡大している．従来の分散データベースシステムでは，転送す

† キヤノン株式会社情報通信システム本部経営情報システムセンター
Management Information Systems Development & Operation Center, Information & Communication Systems Headquarters, Canon Inc.
†† 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
††† 大阪大学サイバーメディアセンター応用情報システム研究部門
Applied Information Systems Division, Cybermedia Center, Osaka University

本論文の内容は 1999 年 11 月マルチメディア通信と分散処理研究会にて報告され，同研究会主査により情報処理学会論文誌への掲載が推薦された論文である．

るデータ量の削減が性能向上の重要な要因であったため，データベースを特定のサイトに固定し，処理依頼と処理結果のメッセージ交換によって処理を行っていた．これに対して，広帯域ネットワークを利用すればデータベース全体の転送も短時間で行えるため，データベースを移動してそれに対する処理をローカルに行うことで通信の回数を減らすことが可能となる．筆者らの研究グループでは，このようなデータベース自体を移動してトランザクション処理を行う手法（データベース移動）を提案した[7]．

さらに，筆者らの研究グループは文献 8) において，移動機能を有する分散データベースシステム DB–MAN を提案した．しかし，文献 8) ではシステムのおおまかな機能について議論しているのみであり，また，実環境に適用できない仮定もいくつかされていた．そこで本論文では，実環境におけるデータベース移動の実現について議論する．まず，データベース移動の機能を有するシステムの詳細な設計を行い，この設計に基づいて実装したプロトタイプシステム DB–MANα について述べる．次に，システムの性能評価のために行った実測評価の結果を示す．

DB–MANα システムでは，データベース移動を用いた処理（移動処理）が従来のデータベース固定型の処理（固定処理）に比べてつねに処理時間が短いとは限らないことを考慮して筆者らが提案した，固定処理と移動処理を適応的に選択する手法[1]を用いてトランザクションを実行する．また，DB–MANα システムにおいて，データベースをディスク上に記録すると，ディスクアクセスがボトルネックとなりデータベース移動に大きな時間がかかってしまう．近年の主記憶の価格低下にともない，主記憶データベース[4),5)]を現実的に利用できるようになってきたことから，DB–MANα では主記憶データベースを用いることで，データベース移動を高速化する．

主記憶データベースでは一般にディスク上のバックアップなどを用いて障害に対処するが[6),11)]，提案システムではデータベースが移動することから，一般的なバックアップやログ情報の記録方法をそのまま適用することはできない．そこで，DB–MANα システムでは，移動を考慮したバックアップ管理手法を導入する．

以下，3 章でシステムの設計について述べ，4 章でシステムの実装について述べる．さらに 5 章で実装したシステムの評価を行い，最後に 6 章で本論文のまとめと今後の課題について述べる．

2. 関 連 研 究

分散処理においてデータベースやデータ項目の移動を考えた研究はいくつか報告されている．最も一般的なものは，ネットワークを介して相互接続されているサーバ間の負荷を均一化することを目的としてデータの配置を再編成するものである．文献 3)，10)，12)〜14)，18) などがそれにあたる．しかし，これらの研究は広帯域ネットワークを想定しておらず，データの移動は大きなオーバヘッドとなると考えている．したがって，データの移動を頻繁には行わず，本研究のようにトランザクション単位でデータベースを移動するものとは仮定が異なる．

本研究同様に，広帯域ネットワークを想定してデータベースを移動することを考えた研究としては，文献 2)，9) などがある．文献 9) では，分散したサイトにデータベースを定期的にブロードキャストする datacycle と呼ぶ手法を提案して，読出し操作のみのトランザクションが中心的な環境で高いスループットを実現している．しかし，この手法では，書込み操作は特定の 1 つのサイトで行っているので，本質的には分散データベースではない．また，リング型ネットワーク上のみに特化した手法であるため，その他のトポロジのネットワーク上では実現できない．文献 2) では，トランザクション処理に必要なデータ項目をトランザクション発生サイトに転送する send-on-demand という本論文のデータベース移動に相当する手法を提案している．この手法は，datacycle が書込み操作が中心的なトランザクションにおいて処理効率が低下することに着目し，このようなトランザクションにおいて高い処理効率を実現するために提案されたものである．さらに文献 2) では，読出し操作のみのトランザクションでは datacycle を用い，書込み操作をともなうトランザクションでは send-on-demand を用いるといった混合型の手法を提案し，datacycle と send-on-demand の両者の長所を生かしてトランザクション処理効率を向上することを図っている．しかし，これらの手法では，datacycle を基盤としているため，datacycle と同様にリング型のネットワーク上でしか利用できない．また，すべての書込み操作をトランザクション発生サイトへデータ項目を移動することで処理しているため，転送すべきデータ項目の総サイズが大きくなると処理効率が劣化する．したがって，多くのデータベースがかかわるような複雑な処理を必要とするトランザクションが頻繁に発生するような環境では，効率的に動作しない．

3. システム設計上の問題点とその解決方法

本章では，筆者らの研究グループが文献 8) で提案した，DB–MAN システムを実環境に適用するうえで問題となる点と，その解決方法について述べる．

3.1 トランザクション処理手法の選択

DB–MAN システムでは，トランザクショ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に基づいて，トランザクション処理手法として移動処理もしくは固定処理を選択する．しかし，文献 8) 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，DB–MAN システムの手法選択においては，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．

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，DB–MAN システムで用いていたトランザクション処理手法の選択方法をそのまま適用することはできない．したがって，本論文において構築する DB–MANα システムでは，将来到着するトランザクション系列を予測してトランザクション処理手法を選択する必要がある．

筆者らの研究グループでは，文献 1) において，アクセスパターンが既知でない環境を想定し，到着するトランザクション系列を予測してトランザクション処理手法を選択する方法（連続系列手法）を提案している．DB–MANα システムでは，この連続系列手法を，トランザクション処理手法の選択法として用いる．以下では，文献 1) で提案した，連続系列手法について説明する．

[連続系列手法]

データベース移動が有効となるのは，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場合など，特定のサイトからアクセスが集中して発生する場合である．このようなアクセスの偏りを検出するために，各データベースに対する次の 3 つのアクセス情報を記録する．

S： トランザクション系列の発生サイト

P_A： トランザクション系列がアクセスしたデータページ数の累計

Q： トランザクション系列に含まれる問合せの数

P_A および Q は，トランザクションが同一のサイトから発生している限り加算され，そのサイトからの最近のアクセスの累計となる．トランザクションが，記録されているサイトと異なるサイトから発生した場合には S，P_A および Q のすべての情報が新たなサイトからのアクセス情報で置き換えられる．なお，このアクセス情報の更新は，トランザクション終了時に行う．

このように記録したアクセス情報を用いて，固定処理と移動処理のどちらを用いるかをトランザクション開始時に選択する．具体的には，次のすべての条件が満たされるときに移動処理を選択する．

(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が自サイトにない．

(2) そのデータベースのアクセス情報に記録されているトランザクション系列の発生サイトが，到着したトランザクションの発生サイトと同一である．

(3) そのデータベースのアクセス情報に記録されているトランザクション系列を，固定処理によって処理した場合の通信所要時間の推定値 T_{fix} と，移動処理によって処理した場合の通信所要時間の推定値 T_{mig} が次の条件を満たす．

$$T_{fix} > T_{mig}$$

ここで，通信所要時間の推定値 T_{fix} および T_{mig} は，S，P_A，および Q と，注目しているデータベースの総ページ数 P_{DB}，1 ページ分の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_T，および，データがネットワーク内を伝わる遅延（伝播遅延）D_P から，それぞれ次式のように算出する．

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

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

T_{fix} の第 1 項は，問合せ結果の返送のためのデータ転送遅延を表しており，第 2 項は，問合せおよびその結果の返送のための伝播遅延を表している．また，T_{mig} の第 1 項は対象となるデータベースの転送にかかる遅延を表しており，第 2 項は，移動要求，データベース移動および移動完了通知のための通信の伝播遅延を表している．本研究で想定する広帯域ネットワークにおいては，T_{mig} は第 2 項が，T_{mig} は第 1 項が支配的となる．ここで，P_A はトランザクションがアクセスしたデータ量であるため，固定処理において結果として返されるデータ量は P_A よりも少なくなることがある．この場合，推定値と実際の値に誤差が生じてしまうが，T_{fix} においては第 2 項が支配的となるため，その影響は比較的小さいものと考えられる．

このように，連続系列手法では，特定のサイトから連続してトランザクションが発生していた場合，集中したアクセスが発生していると見なし，さらに連続したアクセスが発生するものと予想してデータベースを移動する．

3.2 バックアップ管理

移動処理の有効性を高めるためには，データベース

移動をできるだけ高速に行う必要がある．DB–MANシステムでは，データベース移動を高速化するためにデータベースを主記憶データベースとしていた．

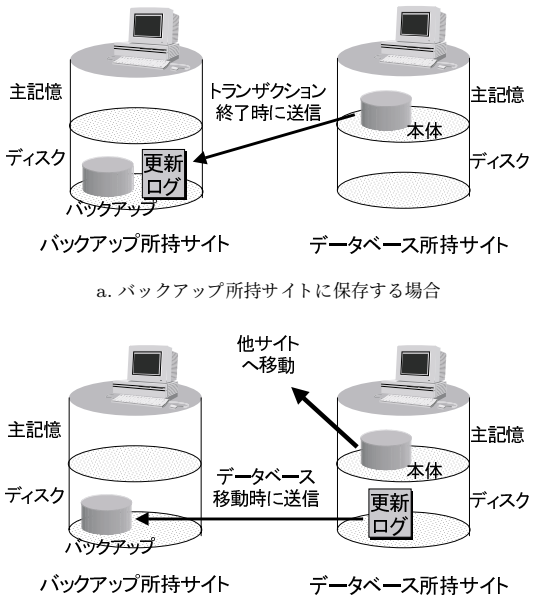
一般に，主記憶データベースでは主記憶の揮発性を考慮して，不揮発性の二次記憶にデータベースのバックアップが置かれる．しかし，移動機能を有する分散データベースシステムでは，データベース本体がサイト間を移動するため，従来のバックアップ管理方法をそのまま用いることはできない．そこで DB–MANα システムでは，移動を考慮したバックアップ管理方法を導入する．

まず，データベースが移動するたびにそのバックアップも移動していたのでは移動時のディスクアクセスの回数が大きくなり，データベース移動に時間がかかるため，バックアップは特定のサイトに固定する．各データベースを作成したサイトにそのデータベースのバックアップを置くものとし，このサイトを**バックアップ所持サイト**と呼ぶ．

データベースを所持するサイト（以下では**データベース所持サイト**と呼ぶ）で何らかの障害が発生し，復旧処理を行う必要が生じたとき，データベース所持サイトはバックアップ所持サイトに対してバックアップの送信要求を出す．バックアップの送信要求を受けたバックアップ所持サイトでは，ディスクに保存されているバックアップから最新のデータベースを取り出し，要求を出したサイトへ送信する．

この処理を行えるようにするためには，バックアップ所持サイトにあるバックアップをつねに最新に保つ必要があるが，データベースを更新したトランザクションが終了するたびにバックアップ所持サイトでバックアップを更新すると，通信やディスクアクセスの回数が増大する．そこで，トランザクションのコミット時にデータベースに対する更新操作のログを作成し，これをディスクに保存する方法を用いる．また，アボート時にはバックアップに関する処理は行わない．この方法を用いることで，いくつかの更新操作が反映される前の古いバックアップと最新の更新操作の内容が記述されたログから，最新のデータベースを作成できるようになる．このログのことを以下では**更新ログ**と呼ぶ．

ここで，更新ログを保存する場所として，バックアップ所持サイトのディスクと，データベースに更新操作を行ったサイト（データベース所持サイト）のディスクの 2 通りが考えられる．バックアップ所持サイトに保存する場合には，**図 1** a に示すように，更新操作を行ったトランザクションがコミットするときにバックアップ所持サイトへ更新操作のログを送信する．データベース所持サイトに保存する場合には，図 1 b に示すように，更新操作を行ったトランザクションがコミットするときに自サイトのディスクに書き込み，データベースの移動時にバックアップ所持サイトに送信する．



a. バックアップ所持サイトに保存する場合

b. データベース所持サイトに保存する場合

図 1 更新ログの保存場所
Fig. 1 Location of the update log.

更新ログをバックアップ所持サイトに保存する場合（図 1 a），データベースの移動時にバックアップ所持サイトへ更新ログを送信する必要がないため，データベース移動にかかる時間が短くなる．その反面，トランザクションのコミット時にバックアップ所持サイトとの通信が必要となるため，コミットにかかる時間が長くなる．更新ログをデータベース所持サイトに保存する場合（図 1 b）には，コミットにかかる時間を短くできる反面，データベース移動にかかる時間が長くなる．

更新ログの保存場所は，システムの利用形態に応じてこれらの 2 つのいずれかをあらかじめ選択する．

4. システムの設計と実装

本章では，前章の議論に基づく DB–MANα システムの設計と実装について述べる．

4.1 システムの設計

DB–MANα システムは対称型のシステムであり，すべてのサイトに同一のサブシステムが存在する．サブシステムのシステム構成を**図 2** に示す．図中のモ

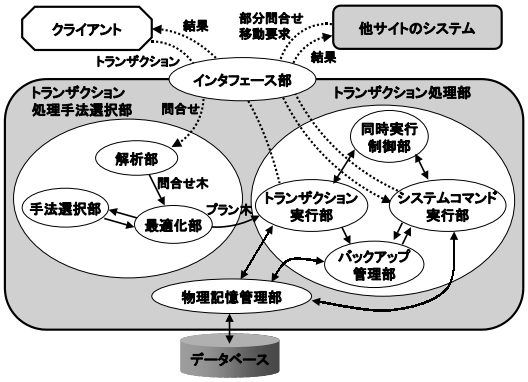


図 2 システム構成
Fig. 2 System composition.

ジュールのうち，手法選択部とバックアップ管理部は，前章で述べた DB–MAN システムの問題点の解決法を実現するモジュールである．その他のモジュールは，従来の分散データベースとほぼ同様のものである．以下では，図中の各モジュールの機能について簡単に述べる．

インタフェース部： クライアントからトランザクションを受け取り，トランザクション処理手法選択部に問合せを 1 つずつ渡す．一方，部分問合せや移動要求など，他サイトのシステムから受け取った処理要求は，直接トランザクション処理部に渡す．

トランザクション処理手法選択部： トランザクション処理手法選択部は，次の 3 つの各部からなる．

解析部： クライアントから文字列で渡された問合せを解析し，問合せ木を構成して最適化部に渡す．

最適化部： 問合せ木を，実際のデータベース操作を記述したプラン木に変換し，トランザクション処理部へ渡す．

手法選択部： 3.1 節で述べた手法に基づいて，固定処理と移動処理のどちらで処理を行うかを決定する．

トランザクション処理部： トランザクション処理部は，次の 4 つの各部からなる．

トランザクション実行部： トランザクション処理手法選択部から，プラン木と処理手法の情報を受け取り，指定された処理手法でプラン木を実行する．

システムコマンド実行部： 自サイトおよび他サイトのトランザクション実行部およびシステムコマンド実行部から，データベース移動，部分問合せの実行，カタログ情報の更新などのシステムによる処理要求（システムコマンド）を受け取り，実行する．

同時実行制御部： データベース単位の 2 相施錠規約に基づいて，複数のトランザクションまたはシステムコマンドの同時実行を制御する．

バックアップ管理部： 3.2 節で述べた方法で，バックアップを管理する．更新ログは，データベース移動にかかる時間を短くすることを重視し，バックアップ所持サイトに保存するとした．

物理記憶管理部： トランザクション処理部からの要求に応じて，物理記憶に対しデータの入出力を行う．

4.2 システムの実装

実装は，カリフォルニア大学バークレー校で開発されたアカデミックフリーのリレーショナルデータベースシステムである POSTGRES [16),17)]をベースに，次の 3 つの部分を中心に行った．

- システムの分散化
  - システム間の通信管理機能
  - 問合せを部分問合せに分割する機能
  - 2 相コミット機能
- データベース移動機能の追加
  - データベースの位置情報管理機能
  - トランザクション処理手法選択機能
  - インデックスの移動機能
  - リレーションログの導入
- 主記憶データベース化
  - 物理記憶管理部の変更
  - バックアップ管理機能

実装は C 言語で行い，上記の機能を合計して約 30,000 行を POSTGRES に追加した．なお，システムの分散化においては，POSTGRES と同じくカリフォルニア大学バークレー校で開発された分散データベースシステムである Mariposa [15)]のコードを一部流用した．また，POSTGRES はリレーショナルデータベースであるため，データベース移動の単位はリレーションとして実装した．

以下では，移動後の処理を高速化するために実装において導入した，インデックスの移動とリレーションログについてそれぞれ 4.2.1 項および 4.2.2 項で述べる．

4.2.1 インデックスの移動

移動後の処理を高速化するためには，インデックスをリレーションとともに移動させる必要がある．

POSTGRES では，インデックスはリレーションとして実装されており，タップルの識別にはページ番号とページ内オフセットが用いられているため，インデックスリレーションをそのまま移動すれば移動先でもそのインデックスは機能する．また，システムがインデックスを認識するためには，システムカタログにインデックスの情報が登録されていなければならないが，DB–MANα システムの実装ではインデックスの作成時にその情報をブロードキャストし，全サイトのシステムカタログに登録するものとした．

4.2.2 リレーションログの導入

POSTGRES では，ログとしてそれぞれのトランザクションがコミットしたかアボートしたかという情報しか記録されない．トランザクションが行った更新操作などの内容は，データベース内の各タップルに，それを挿入したトランザクションの ID と削除したトランザクションの ID をそれぞれ書き込むことによって保持している．タップルに書かれたトランザクション ID を持つトランザクションがコミットされていればその挿入操作は有効，アボートされていれば無効であると判断する．タップルを削除するにはそのタップルに削除したトランザクションの ID を書き，更新のときは古いタップルを削除して新しいタップルをリレーションの最後に追加する．トランザクションのロールバック時には，ログにそのトランザクションがアボートしたと書けばすべての更新操作が無効になる．

ここで，もともとの POSTGRES は分散データベースではないため，ログはシステムが動作しているサイトに 1 つしか存在しない．システムはリレーションからタップルを 1 つ取り出すたびに，ログを見てそのタップルの有効性を確認しなければならないが，そのリレーションが移動してきたものである場合，そのサイトにログはない．そこで DB–MANα システムの実装では，ログをリレーションごとに作成し（これを**リレーションログ**と呼ぶ），リレーションの移動時に一緒に移動するものとした．

5. 実 測 評 価

本章では，実装した DB–MANα システムの性能評価のために行った実験とその結果について述べ，さらに結果について考察する．

5.1 実 験 環 境

実験では，DB–MANα システムのトランザクション処理にかかる平均応答時間を測定し，固定処理のみで処理した場合，および，移動処理のみで処理した場合と比較した．

表 1 実験環境およびパラメータ
Table 1 System environment and parameter configuration.

ネットワーク環境	
サイト数	3
実効帯域幅	約 80 [Mbps]
伝播遅延*	200 [ミリ秒] （50〜400 [ミリ秒]）
データベース	
データベース（リレーション）数	1
データベースサイズ*	31.5 [MB] （3.5〜94.5 [MB]）
トランザクション	
問合せ/トランザクション*	10（1〜30）
返送するデータ量*	1000 [タプル] （600〜15000 [タプル]）
到着間隔変更周期*	400 [秒]（30〜1600 [秒]）
到着間隔比*	5（1〜20）
– 集中時平均到着間隔	30 [秒]
– 散発時平均到着間隔	150 [秒]（30〜600 [秒]）

実験を行ったシステム環境と，各実験で用いたパラメータを**表 1** に示す．実験は，表中の * 印をつけたパラメータについて，それぞれ括弧内に示した範囲で値を変化させて行った．変化させるパラメータ以外は括弧外の値に固定した．

すべての実験で，サイト数は 3 に，データベース（リレーション）数は 1 に固定した．これは，DB–MANα の手法選択機構では，選択条件はサイト数に依存せず，データベースごとに独立して処理手法を選択するため，これらのパラメータを変化させても結果に影響しないものと考えられるからである．ネットワークは，100 Mbps のイーサネットを用いており，その実効帯域幅は約 80 Mbps である．また，伝播遅延を変化させる実験を行うため，伝播遅延はプログラム内で擬似的に発生させた．

トランザクションは，サイトごとに平均到着間隔をパラメータとして与え，指数分布に基づいて発生させた．特定サイトからの集中的なアクセスは，ある 1 つのサイトの平均到着間隔を，それ以外のサイトの平均到着間隔に比べて小さな値に設定することで発生させた．集中してアクセスが発生するサイトの平均到着間隔を集中時平均到着間隔，それ以外のサイトの平均到着間隔を散発時平均到着間隔と呼ぶ．集中的にデータベースにアクセスするサイトは一様分布の乱数によって決定し，到着間隔変更周期ごとにそのサイトを変化させた．また，到着間隔比は (散発時平均到着間隔)/(集中時平均到着間隔) で与えられる数値で，この値が大きいほど，特定サイトから集中的にアクセスが発生していることを表す．各トランザクションには，イン

デックスを作成している属性に対して，その値を指定する検索問合せが含まれる．データベース作成時に，インデックスを付加する属性に対応して，規則的な数のタップルを作成することで，問合せの結果として取り出すタップルの数を調整した．

DB–MANα システムでは，データベース移動と読み書きなどのデータベース操作の同時実行制御機構が実装されていないため，移動中のデータベースに対してデータベース操作を実行できない．したがって，データベースの移動中に他のトランザクションが到着すると，そのトランザクションはデータベース移動が終了するまで待機しなければならない．そこで実験では，トランザクションの同時実行は行わず，直前のトランザクションが終了してから次のトランザクションの処理を開始した．このとき，実装した手法選択機構の有効性を検証することを目的として，未実装の機能による不当な処理遅延の影響を防ぐために，直前のトランザクションの終了までの待ち時間は応答時間に含めないものとした．なお，データベース移動と他のデータベース操作との同時実行制御機構は，今後実装する予定である．

以下の実験結果のグラフでは，次の表記を用いる．

DB–MANα： DB–MANα システムの手法選択機構を用いて，トランザクション処理方法を適応的に選択した場合の平均応答時間．

MIG： すべてのトランザクションを，移動処理のみで処理した場合の平均応答時間．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： 従来の分散データベースシステムと同様に，すべてのトランザクションを，固定処理のみで処理した場合の平均応答時間．

5.2 実験 1：データベースサイズの影響

データベースのサイズを変化させたときの平均応答時間を**図 3** に示す．

データベースサイズが小さいときには，データベースの移動にかかる時間が短いため，固定処理に比べて通信回数が少ない移動処理が良い結果を示す．DB–MANα では，ほとんどのトランザクションで移動処理を選択するため，移動処理のみの場合の性能と近くなる．しかし，DB–MANα では，同じサイトから連続してトランザクションが発生した場合のみ移動処理を選択するため，トランザクション発生サイトが変化した直後のトランザクションでは必ず固定処理を選択する．そのため，データベースサイズが非常に小さいとき，移動処理のみの場合より平均応答時間が少し長くなる．

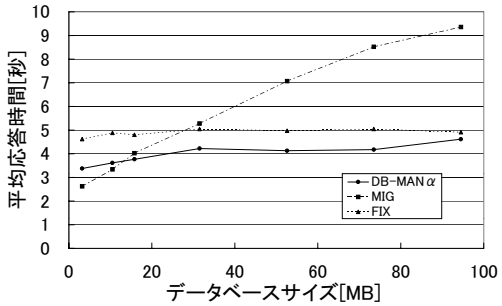


図 3 データベースサイズと平均応答時間
Fig. 3 Database size and average response time.

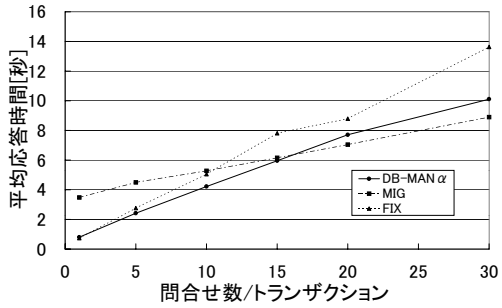


図 4 問合せ数と平均応答時間
Fig. 4 Number of queries and average response time.

データベースサイズが大きいときは，データベース移動にかかる時間が大きいため，移動処理より固定処理が良い結果を示す．DB–MANα では，データベースサイズが大きくなるとほとんどのトランザクションで固定処理を選択するため，固定処理のみの場合の性能と近くなる．

全般的には，DB–MANα は，データベースサイズを考慮して処理手法を選択するため，データベースサイズにかかわらず良い結果を示している．

5.3 実験 2：トランザクションに含まれる問合せ数の影響

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平均応答時間を**図 4** に示す．

DB–MANα では，トランザクションに含まれる問合せ数が少ないとき，特定のサイトでアクセスが集中していると判断されないため，ほとんどのトランザクションで固定処理を選択する．このため，DB–MANα と固定処理のみの場合の結果が近い値を示している．しかし，稀に特定サイトから長期にわたって集中的なトランザクションが発生することにより，あまり有効でないデータベース移動を行うため，トランザク

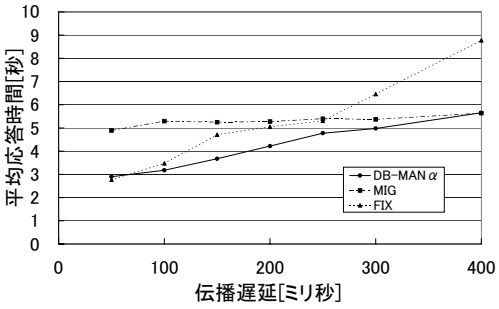


図 5 伝播遅延と平均応答時間
Fig. 5 Propagation delay and average response time.

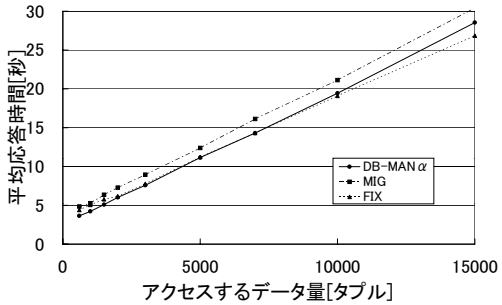


図 6 問合せで返送されるデータ量と平均応答時間
Fig. 6 Amount of data returned to a query and average response time.

ションに含まれる問合せ数がきわめて少ないときには，DB–MANα は固定処理のみの場合よりわずかに悪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多いときは，固定処理では通信回数が多くなるため，移動処理が良い結果を示す．このとき，DB–MANα では，ほとんどのトランザクションで移動処理を選択するため，移動処理のみの場合に近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非常に多い場合には，つねに移動処理を行った方が有効であるため，アクセス集中時にのみ移動処理を行う DB–MANα が移動処理のみの場合よりも悪い結果を示している．しかし，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から発生するバッチ処理である場合が多いため，手法選択機構にトランザクションを発生したアプリケーションを判別する機能を持たせることで，性能を改善できるものと考えられる．

5.4 実験 3：伝播遅延の影響

伝播遅延を変化させたときの結果を**図 5** に示す．

伝播遅延が小さいとき，通信回数は処理時間にそれほど影響しないため，移動処理より固定処理が良い結果を示す．このとき，DB–MANα はほとんどのトランザクションで固定処理を選択するが，実験 2 の場合と同様に，稀に長期にわたって集中的なトランザクションが発生することによってあまり有効でないデータベース移動を行うため，伝播遅延が非常に小さいときには，固定処理のみの場合よりもわずかに悪い結果を示している．

伝播遅延が大きいときは，通信回数が処理時間に大きな影響を与えるため，固定処理より移動処理が良い結果を示している．このような場合，DB–MANα では移動処理が選択されることが多くなるため，移動処理のみの場合に近い結果となる．また，伝播遅延が非常に大きくなると，つねに移動処理を行った方が有効であるため，DB–MANα よりも移動処理のみの場合の方がわずかに良い結果を示す．しかし，実環境において，伝播遅延が 400 ミリ秒を超えるようなことは稀なので，DB–MANα はほとんどの大きさの伝播遅延に対して有効である．

5.5 実験 4：各問合せで返送されるデータ量の影響

1 つの問合せで返送されるデータ量を変化させたときの平均応答時間を**図 6** に示す．

返送されるデータ量が少ないときは，固定処理において問合せ結果を転送する時間が短くて済むため，移動処理のみの場合よりも固定処理のみの場合が良い結果を示している．また，このような場合には，DB–MANα では固定処理が多く選択されるようになる．さらに，移動処理の処理時間とリモートサイトに対する固定処理の処理時間との差が小さいため，返送されるデータ量が非常に小さい場合も，散発的なデータベース移動による結果への影響は小さく，DB–MANα は最も良い結果を示している．

返送されるデータ量が多くなるにつれて，固定処理における問合せ結果のデータ転送量は増加するが，平均応答時間は移動処理に比べて短くなっている．これは，各サイトにおける問合せの処理，および，リモートサイトとトランザクション発生サイト間のデータの受け渡しを並行に実行できるためである．つまり，リモートサイトから結果を転送している間に，トランザクション発生サイトで処理を実行でき，また，リモートサイトで部分問合せを実行している間に，トランザクション発生サイトでは，すでに受け取った結果の処理を行える．このようなことから，固定処理が移動処理よりも良い結果を示している．

各サイトでの処理や結果の転送は非同期に実行されるため，処理時間の見積りが難しいことから，DB–MANα では，それらが直列に処理される場合の処理時

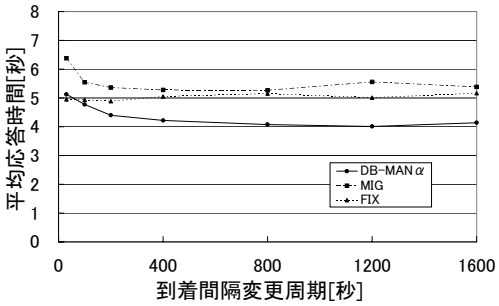


図 7 到着間隔変更周期と平均応答時間
Fig. 7 Period to change the arriving interval and average response time.

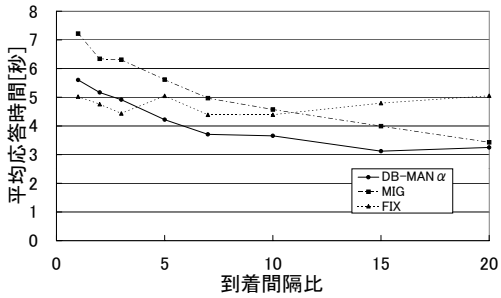


図 8 到着間隔比と平均応答時間
Fig. 8 Ratio of arriving interval and average response time.

間を見積り値として計算し手法の選択を行う．そのため，アクセスするデータ量が多くなると，DB–MANα では，移動処理を選択しやすくなることから，固定処理のみの場合よりも悪い結果を示している．したがって，返送されるデータ量が多くなる場合には，各サイトでの処理や結果の転送などの非同期な実行を考慮するように，手法選択機構を拡張する必要がある．

5.6 実験 5：到着間隔変更周期の影響

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平均応答時間を**図 7** に示す．

到着間隔変更周期が短いときには，特定サイトからの集中的なアクセスがそれほど継続しないため，移動処理より固定処理が良い結果を示す．このとき，DB–MANα ではほとんどのトランザクションで固定処理を選択する．実験 2 および実験 3 の場合と同様に，稀に起こる長期にわたる集中的なトランザクションの発生により，有効でないデータベース移動が行われるため，到着間隔変更周期が非常に短いときには，固定処理のみの場合の方がわずかに良い結果を示している．

到着間隔変更周期が長くなるにつれて，特定サイトからのアクセスが長くなるため，DB–MANα と移動処理のみの場合の平均応答時間が短くなる．しかし，移動処理のみの場合では，集中的なアクセスの間に他サイトから散発的に発生するトランザクションよってあまり有効でないデータベース移動を行うため，平均応答時間は固定処理のみの場合よりも長くなる．

DB–MANα は，アクセスの集中する度合いを考慮して適応的にデータベース移動を行うので，ほとんどの場合で良い結果を示している．

5.7 実験 6：到着間隔比の影響

トランザクションの到着間隔比を変化させたときの平均応答時間を**図 8** に示す．

到着間隔比が小さいときは，特定サイトからの集中したアクセスがほとんど発生しないため，移動処理より固定処理が良い結果を示す．このとき，DB–MANα はほとんどのトランザクションで固定処理を選択するが，実験 2，実験 3，および，実験 5 の場合と同様に，稀に起こる集中的なトランザクションの発生によって有効でないデータベース移動が行われるため，固定処理のみの場合よりもわずかに悪い結果を示している．

到着間隔比が大きいとき，すなわち特定サイトからアクセスが集中するときは，固定処理より移動処理が良い結果を示す．到着間隔比が大きくなるにつれて，DB–MANα はほとんどのトランザクションで移動処理を選択するようになるため，移動処理のみの場合に近い結果を示している．

全般的には，DB–MANα は到着間隔比によらず良い結果を示している．アクセスが極端に集中する場合や，まったく集中しない場合でも，従来の固定処理のみの場合や移動処理のみの場合とほぼ同じ性能を示す．

6. おわりに

本論文では，移動機能を有する分散データベースシステムを設計し，そのプロトタイプシステム DB–MANα を実装した．DB–MANα システムは，トランザクション処理手法を適応的に選択する機構と，データベース移動を考慮したバックアップ管理機構を有する．また，データベースが移動した後の処理を高速化するために，インデックスを移動する機能を実装し，リレーションログを新たに導入した．

さらに，DB–MANα システムの実測評価により，適応的なトランザクション処理手法選択機構の有効性を検証した．これにより，DB–MANα システムは従来の分散データベースシステムより処理時間を短縮できることを確認した．特に，特定のサイトから連続したアクセスがあるとき，DB–MANα によって処理時間

が大幅に改善されることが分かった．

今後の研究課題としては，リモート処理における，各サイトでの処理や結果の転送などの非同期実行を考慮するように，DB–MANα システムの手法選択機構を拡張することが必要である．非同期実行によるリモート処理の高速化を考慮した手法選択機構が実現すれば，より高精度に適切なタイミングでデータベースを移動できるようになり，DB–MANα システムの性能が向上するものと考えられる．また，文献 1) の手法では，現在連続してデータベースにアクセスしているサイトのみの情報を記録して移動を決定するため，複数のサイトが同時にトランザクションを発生するような環境では，移動がほとんど起こらない．今後，このような環境に適した手法選択法についても検討する必要がある．

さらに，移動中のデータベースに対するデータベース操作をサポートする同時実行制御機構についても，今後実装する予定である．

謝辞 本研究は，日本学術振興会研究費奨励研究 (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) 秋山豊和，原 隆浩，春本 要，塚本昌彦，西尾章治郎：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，情報処理学会論文誌，Vol.40, No.6, pp.2765–2775 (1999).
2) Banerjee, S., Li, V.O.K. and Wang, C.: Distributed database systems in high-speed wide area networks, *IEEE Journal on Selected Areas in Communications*, Vol.11, No.4, pp.617–630 (1993).
3) Devine, R.: Design and implementation of DDH: A distributed dynamic hashing algorithm, *Proc. 4th Int'l Conf. on Foundations of Data Organization and Algorithms* (*FODO*), pp.101–114 (1993).
4) 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 (1984).
5) Garcia-Molina, H., Lipton, R.J. and Valdes, J.: A massive memory machine, *IEEE Trans. Comput.*, Vol.C-33, pp.391–399 (1984).
6) Hagmann, R.: A crash recovery scheme for a memory resident database system, *IEEE Trans. Comput.*, Vol.C-35, pp.839–843 (1986).
7) Hara, T., Harumoto, K., Tsukamoto, M. and Nishio, S.: Database migration: A new architecture for transaction processing in broadband networks, *IEEE Trans. Knowledge and Data Eng.*, Vol.10, No.5, pp.839–854 (1998).
8) Hara, T., Harumoto, K., Tsukamoto, M. and Nishio, S.: DB–MAN: A distributed database system based on database migration in ATM networks, *Proc. IEEE Data Engineering.*, pp.522–531 (1998).
9) Herman, G., Gopal, G., Lee, K. and Weinrib, A.: The datacycle architecture for very hight throughput database systems, *Proc. ACM SIGMOD '87*, pp.97–103 (1987).
10) Johnson, T. and Krishna, P.: Lazy updates for distributed search structure, *Proc. ACM SIGMOD '93*, pp.337–346 (1993).
11) Lehman, T.J. and Carey, M.J.: A recovery algorithm for a high performance memory resident database system, *Proc. ACM SIGMOD '87*, pp.104–117 (1987).
12) Litwin, W., Neimat, M.-A. and Schneider, D.A.: LH* – linear hashing for distributed files, *Proc. ACM SIGMOD '93*, pp.327–336 (1993).
13) Matsliach, G. and Shmueli, O.: An efficient method for distributing search structures, *Proc. 1st Int'l Conf. on Parallel and Distributed Information Systems* (*PDIS*) (1991).
14) Severance, C., Pramanik, S. and Wolberg, P.: Distributed Linear Hashing and Parallel Projection in Main Memory Databases, *Proc. 16th International Conference on Very Large Data Bases*, Brisbane, Queensland, McLeod, D., Sacks-Davis, R. and Schek, H.-J. (Eds.), pp.674–682, Morgan Kaufmann (1990).
15) Stonebraker, M., Aoki, P.M., Devine, R., Litwin, W. and Olson, M.: Mariposa: A new architecture for distributed data, *Proc. IEEE Data Engineering*, pp.54–65 (1994).
16) Stonebraker, M. and Rowe, L.A.: The design of POSTGRES, *Proc. ACM SIGMOD '86*, pp.340–355 (1986).
17) Stonebraker, M., Rowe, L.A. and Hirohama, M.: The implementation of POSTGRES, *IEEE Trans. Knowledge and Data Eng.*, Vol.2, No.1, pp.125–142 (1990).
18) Vingralek, R., Breitbart, Y. and Weikum, G.: Distributed file organization with scalable cost/performance, *Proc. ACM SIGMOD '94*, pp.253–264 (1994).

(平成 12 年 2 月 14 日受付)
(平成 12 年 9 月 7 日採録)

推　薦　文

データベース移動のコンセプトに基づき，トランザクションの処理方法を適応的に選択する手法を提案し，これを実装したシステムの性能を実測評価し，良好な結果を確認している．ネットワークの帯域幅が大きくなる将来の通信環境へ向けての新手法の具体的な提案であり，基礎的な技術検証として評価できる．また，今後の研究の広がりも期待できる．
（マルチメディア通信と分散処理研究会主査　滝沢誠）

酒井　　仁

1998年大阪大学工学部情報システム工学科卒業．2000年同大学院工学研究科修士課程修了．同年，キヤノン（株）入社，現在に至る．データベースシステムに興味を持つ．

海貝　明道

1999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．次世代データベースに興味を持つ．

秋山　豊和（正会員）

1997年大阪大学工学部情報システム工学科卒業．1999年同大学院工学研究科博士前期課程修了．2000年同大学院工学研究科博士後期課程中退後，大阪大学サイバーメディアセンター助手となり，現在に至る．分散処理，データベースに興味を持つ．IEEE，電子情報通信学会各会員．

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．工学博士．1996年本学会山下記念研究賞受賞．2000年電気通信普及財団テレコムシステム技術賞受賞．分散データベースシステム，アドホックネットワークに興味を持つ．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ー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．2000年より大阪大学サイバーメディアセンター長を併任．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．