データベースシステム 113－18
（1997. 7. 14）

データベース移動に基づく動的複製配置法

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
〒565 吹田市山田丘 2-1
{hara,harumoto,tuka,nishio}@ise.eng.osaka-u.ac.jp

近年，ATM 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激に拡大している．このような広帯域ネットワークでは，データベース全体の移動も短時間で行うことができる．筆者らはこの点に着目して，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，シミュレーション評価によってその有効性を確認している．一方，従来の分散データベースでは，処理性能を向上させるために複製を用いることが一般的である．そこで本稿では，提案したトランザクション処理手法に基づいて，データベース移動を利用した動的複製配置法を提案する．更に，提案する動的複製配置法とこれまでに提案した手法をシミュレーションによって比較することで，提案する手法の有効性を検証する．

Dynamic Replica Allocation based on Database Migration

Takahiro HARA　Kaname HARUMOTO　Masahiko TSUKAMOTO　Shojiro NISHIO

Department of Information Systems Engineering, Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565, Japan
{hara,harumoto,tuka,nishio}@ise.eng.osaka-u.ac.jp

As one of the new technologies to make good use of recent broadband networks, dynamic relocation of databases through such high performance networks, which we call *database migration*, will soon become a powerful and basic database operation of practical use. Based on this database migration operation in distributed database environments, we have proposed a transaction processing method so far. In general, replication of databases is one of the effective techniques for improving the transaction processing throughput in conventional systems. In this paper, we propose a replica management method based on the method which we proposed in our previous paper. This method dynamically relocates database replicas using database migration at the beginning of a transaction. We also show simulation results regarding performance evaluation of our proposing method.

1 まえがき

広帯域ネットワークを有効利用して分散処理を高速化する一つの可能性として，データベース移動（以下，DB 移動と呼ぶ）が考えられる [3][5]. 筆者らは，これまでにDB 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 [2][4]. 提案した手法は，トランザクションの処理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，データベースへのアクセスパタンなどを考慮して，従来のデータベース固定型の処理と DB 移動を用いた処理とを適応的に選択するものである.

提案した手法では，DB 移動後の転送元サイトのデータベースを削除することで，データベースの複製を作成しないようにしている. これは，複製を作成すると，複製間の一貫性保持など管理が複雑になり，そのためのメッセージ通信によって性能が低下することを考慮したものである. しかし，複製を作成することで，データベース操作のためのメッセージ通信を削減できることから，従来の分散データベースの分野では複製を用いたトランザクション処理手法が多く提案されている [6][7]. 複製を作成するかしないかは，ネットワークの帯域幅などシステムの特性によって，どちらが有効であるかが決定されるものと考えられる.

そこで本稿では，筆者らが提案した DB 移動を用いたトランザクション処理手法に基づいて，DB 移動を用いて動的に複製を配置してトランザクションを処理する手法を提案する. これまでに提案されている複製を用いたトランザクション処理手法は，ネットワークの帯域幅が狭いことを前提としており，複製を静的に配置したり，ある程度長い観測期間内のアクセス履歴などの統計情報を用いて複製の再配置をするものがほとんどである. 一方，本稿で提案する手法では，豊富な帯域幅を前提とし，DB 移動を用いることでデータベースといった大きなデータ単位の複製の作成・削除をトランザクション毎に動的に決定する. なお本稿では，高速なデータベース・アクセスと DB 移動を実現するために主記憶データベース [1] を想定する. 更に，提案する手法とこれまでに提案したトランザクション処理手法とをシミュレーションによって比較することで，提案する手法の有効性を検証する.

2 DB 移動を用いた動的複製配置法

本章では，DB 移動を用いて複製を動的に配置してトランザクションを処理する手法（DB 移動複製法）を提案する.

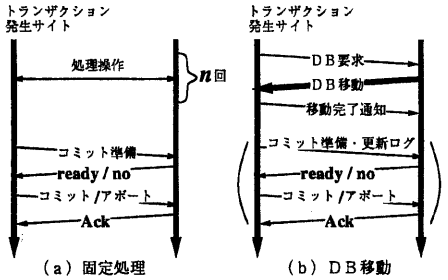


図 1: 各手法の通信手順

2.1 システムの基本動作

本稿では，ATM の仮想LAN などのように，メッセージのブロードキャストが可能な広帯域ネットワーク内での分散データベース処理を想定する. システム全体として一つのデータベースをもち，これを複数のローカル・データベース（以後は，これをデータベースと呼ぶ）に分割して，システム内の各サイトで主記憶上に保持する. 各サイトは，主記憶領域の範囲内で，複製の作成・削除を行うことができる.

システムの基本動作として，データベースの複製が存在する環境において，二相施錠規約と二相コミット規約に基づいて，トランザクションの処理を実行する. データベースに対する読出し・書込みの操作は，いずれの複製に対しても行えるが，書込みの場合は複製間の一貫性を保つためにすべての複製に書込み施錠を行わなければならない.

なお，トランザクションの処理手法としては，従来のデータベース固定型の処理（以後，固定処理と呼ぶ）と DB 移動を用いた処理のいずれかを選択する. 両手法の処理手順の概要を次に示す.

固定処理：固定処理における通信手順を図 1(a) に示す. 処理依頼メッセージの転送と結果の返送といった処理操作を行い，これに n 回の通信を要する. 最後に，二相コミットのために二往復のメッセージ交換を行う.

DB 移動を用いた処理：DB 移動を用いた処理の通信手順を図 1(b) に示す. まず，DB 要求のためのメッセージをブロードキャストする. DB 要求のメッセージをブロードキャストしているのは，全サイトが各データベースの正確な位置情報を保持することを可能にするためである. なお，本稿では，トランザクションの開始時に，トランザクショ

ンの実行に必要なデータベースが同定できるものと仮定している．次に，DB 移動を実行し，移動完了後に移動完了通知をトランザクション発生サイトからブロードキャストする．文献 [2] では複製を許さないために，DB 移動を実行した後に転送元サイトのデータベースは消去していたが，本稿で提案する手法ではこれを消去しない．トランザクション発生サイトにおいてトランザクション処理に必要なデータベース操作を実行した後，操作対象となったデータベースのいずれかに複製が存在する場合は，複製を保持するサイトに更新ログを転送し，固定処理同様に二相コミットを実行する．更新ログは，コミット準備メッセージに付加する．

2.2 DB 移動複製法

DB 移動複製法では，文献 [2] と同様に，従来の固定処理と DB 移動を用いた処理とを適応的に選択して，トランザクション処理を実行する．手法選択においては，現トランザクションを両手法で実行する際に要するそれぞれの通信所要時間と，トランザクション発生サイトのデータベースのアクセスパタンを考慮する．アクセスパタンを予測するために，データベースのアクセス履歴と，あるサイトが特定のデータベースを連続的に使用する場合に行う連続使用宣言の情報を用いる．

ここで，文献 [2] の手法では，データベースを現在所持しているサイトとトランザクション発生サイトのどちらにそのデータベースが置かれている方がシステム全体として効果的かという点が移動の基準になっていた．しかし，移動後に転送元のデータベースが削除されない場合，現在所持しているサイトとの兼ね合いではなく，移動してくるデータベースとその主記憶領域の確保のために削除されるその他のデータベース（複製）のどちらを所持していることが有効であるかという点を基準に DB 移動の実行を決定する必要がある．以下では，このような考えに基づく手法選択アルゴリズムと複製の作成・削除アルゴリズムの詳細を示す．

2.2.1 手法選択アルゴリズム

DB 移動複製法では，固定処理と DB 移動を用いた処理を適応的に選択してトランザクションを実行する．まず，トランザクション処理の実行に必要なデータベースの総サイズが，トランザクション発生サイトの主記憶領域より大きい場合は，強制的に固定処理を実行する．それ以外の場合では，固定処理と DB 移動を用いた処理の通信所要時間と各サイトのデータベースのアクセスパタンを考慮して処理手法を選択する．したがって，各サイトで手法選択に必要な情報として，各データベースのサイズ情報，(複製を含む) 位置情報，連続使用の情報，アクセス履歴の情報を保持する．

ここで，サイト S_I がデータベース D_j を保持する有効性を示す目的関数 $f(S_I, D_j)$ を次のように定義する．

$$f(S_I, D_j) = \alpha \cdot P' \cdot |L| + \sum_{l=1}^{|L|} \{k_l \cdot (|L| + 1 - l)\} \quad (1)$$

$$\alpha = \begin{cases} 1: & S_I \text{が} D_j \text{の連続使用を宣言している} \\ & \text{あるいは 現トランザクションで} \\ & D_j \text{の連続使用を宣言する．} \\ 0: & \text{その他．} \end{cases}$$

P' : 連続使用優先係数

$|L|$: DB のアクセス履歴のサイズ
(トランザクション L 回分の履歴を残す)

$$k_l = \begin{cases} 1: & l \text{回前のトランザクションで} S_I \text{が} \\ & D_j \text{を利用．} \\ 0: & \text{その他．} \end{cases}$$

$f(S_I, D_j)$ は，第一項で D_j を連続的に使用することに対する優先度を表し，第二項で利用頻度に対する優先度を表している．ここで，連続使用優先係数 P' は，連続使用宣言に対しての優先度の度合を表すもので，文献 [2] の連続使用係数 P と区別して P' としている．

次に，DB 移動を実行する有効性，つまり，トランザクション処理に必要なデータベースの複製を作成することの有効性を表す評価値 E を，次のように定義する．

$$E = \sum_{D_j \in \boldsymbol{D}_N} f(S_I, D_j) - \sum_{D_j \in \boldsymbol{D}_O} f(S_I, D_j) \quad (2)$$

ここで $\boldsymbol{D}_N$ は，トランザクション処理の実行に必要な，トランザクション発生サイトが所持していないデータベースの集合を表している．また $\boldsymbol{D}_O$ は，$\boldsymbol{D}_N$ を主記憶上に配置するために削除しなければならないデータベースの集合である．これを現在所持しているデータベースおよび複製の集合から選択するアルゴリズムについては次節で論ずる．

最終的に，発生したトランザクションを，固定処理と DB 移動を用いた処理のどちらで処理するかを，次の規則にしたがって選択する．

```
if  (T_DB − T_fix) − K′ · E < 0
    then        DB 移動を用いた処理
    else        固定処理
```

ここで，T_{DB}, T_{fix}は，それぞれデータベース移動を用いた処理と固定処理の通信所要時間の見積もり値を表している．この値の算出は，基本的に文献 [2] の方法にしたがって行う．ただし，DB 移動を用いた処理を選択した場合にも，複製間の一貫性を保つために二相コミットを行う必要があるため，DB 移動を用いた処理の通信所要時間は，文献 [2] のものよりもコミット準備とコミットのメッセージのブロードキャストとその返答メッセージの通信分だけ大きくなる．また，K'はEの値の優先度を表すもので，この値の設定がシステム全体の性能に大きく影響する．このK'を履歴依存係数と呼ぶ．また，K'としているのは，文献 [2] の履歴依存係数Kと区別するためである．

2.2.2 複製の作成・削除アルゴリズム

本節では，複製作成・削除アルゴリズムについて述べる．まず，DB 移動の実行に必要な領域を確保するために削除しなければならない複製の集合$\boldsymbol{D_O}$の候補を選択する．各サイトにデータベース領域として割り当てられている主記憶領域のサイズをM，そのサイトが所持しているデータベースおよび複製の集合を$\boldsymbol{D_H}$とすると，$\boldsymbol{D_H}$から現トランザクションの実行に必要なデータベースを除いた集合$\boldsymbol{D_C}$のなかから，次式を満たすようになるまで$\boldsymbol{D_O}$に加える複製を選択する．

$$M > |\boldsymbol{D_H}| - |\boldsymbol{D_O}| + |\boldsymbol{D_N}| \qquad (3)$$

$\boldsymbol{D_O}$の選択法としては，次のアルゴリズムを用いる．

1. $\boldsymbol{D_O}$を空とする．
2. $\boldsymbol{D_C}$のうちで複製が他のサイトに存在するものから，$f(S_I, D_j)$ の値が小さい順に$\boldsymbol{D_O}$に加える．
3. 式(3) を満たしていないならば，$\boldsymbol{D_C}$のうちで複製をもたないものから，$f(S_I, D_j)$ の値が小さい順に$\boldsymbol{D_O}$に加える．なお，ここで選択されたデータベースは削除する前に他のサイトに移動する必要があるため，手法選択の際に，この転送に要する時間も考慮する（T_{DB}に加える）．

$\boldsymbol{D_O}$の決定後，前節の手法選択によって DB 移動を用いた処理が選択された場合，$\boldsymbol{D_O}$に含まれる複製およびデータベースを，移動してくるデータベースの主記憶領域を確保するために削除する（複製の削除）．ただし，複製が存在しないデータベースに関しては，各データベースに対して，全サイトに関する $f(S_I, D_j)$ を計算し，その値が最も大きいサイトへデータベースを移動する．そのサイトの主記憶領域に余裕がない場合には，そのサイトがもつ複製の中から $f(S_I, D_j)$ が最も小さいものを削除して移動するデータベースの領域を確保する．また，そのサイトの主記憶領域が，複製をもたないデータベースや実行中のトランザクションに使用されているデータベースによってすべて占有されている場合には，次に $f(S_I, D_j)$ が大きいサイトを候補にする．全サイトに移動先が見つからない場合は，トランザクションの処理を固定処理に変更する．

更に上記の処理と並行して，トランザクション処理の実行に必要なデータベース$\boldsymbol{D_O}$の複製を，他サイトからの DB 移動によって作成する（複製の作成）．

3 シミュレーションによる性能評価

本章では，本稿で提案した DB 移動複製法の性能評価のために行ったシミュレーションの結果を示す．シミュレーションでは，DB 移動複製法と，従来の複製法，筆者らが文献 [2] において提案したトランザクション処理手法の平均通信所要時間を比較する．比較対象とする従来の複製法と文献 [2] の手法の概要を示す．

従来の複製法: 主記憶領域の許す限り，各データベースの複製をランダムに配置する．ただし，配置は静的なもので，再配置や新たな複製の作成，削除は行わない．トランザクションは，固定処理のみで実行される．

文献 [2] の手法: 文献 [2] の履歴統計手法を用いて，固定処理と DB 移動を用いた処理を選択してトランザクションを実行する．DB 移動を用いた手法を選択した場合に，移動してくるデータベースに対して十分な主記憶領域が残っていない場合は，十分な領域が確保できるまで，DB 移動複製法と同様の方法で，$f(S_I, D_j)$ の値の小さい順に選択したデータベースを他サイトに移動する．

3.1 シミュレーション環境

シミュレーションでは，ATM の仮想 LAN 内に構築された分散データベースシステムを想定す

表 1: シミュレーションのためのパラメータの設定

パラメータ	パラメータの意味	値
$\|S\|$	仮想LAN内のサイト数	20（サイト識別子 $S_1, \cdots, S_{20}$）
$\|D\|$	データベースの総数	20（DB-id $D_1, \cdots, D_{20}$）
$Size(D_j)$	各データベースのサイズ	80 [Mbyte]
p_1	トランザクション発生率1	0.025 (S_1,..., S_{10}), 0.05 (S_{11},..., S_{18}), 0.15 (S_{19}), 0.3 (S_{20})
p_2	トランザクション発生率2	0.025 (S_{11},..., S_{20}), 0.05 (S_3,..., S_{10}), 0.15 (S_2), 0.3 (S_1)
d_{MCS}	MCSへの平均伝搬遅延	0.12 [秒]
d_m	その他の2サイト間の伝搬遅延	0.12 [秒]
C	コネクション設定時間	0.3 [秒]
n	固定処理における通信回数	n' $\|D_N\|$/$\|D_U\|$（n'は1〜30の範囲でランダムに決定する）
$\|L\|$	利用履歴のサイズ	20
P, P'	連続使用優先係数	3.50 (P), 1.40 (P')
K, K'	履歴依存係数	0.02 (K), 0.035 (K')

る．ATMの仮想LANでは，マルチキャストサーバ (MCS) を経由してメッセージをブロードキャストできる．ブロードキャスト以外の通信に関しては，すべてポイント・ツー・ポイントのSVCコネクションを用いる．

シミュレーションで用いるパラメータの値を表1に示す．データベース D_i の初期位置を S_i とする ($1 \leq i \leq 20$)．なお，複製が存在する手法では，複製の初期位置として，主記憶領域が許す限り複製をランダムに配置する．操作回数 n' を1〜30でランダムに変化させることで，様々な複雑度のトランザクションを表現できる．更に，1000回毎に各サイトのトランザクション発生確率を p_1, p_2 と交互に変化させることで，提案した手法の環境の変化に対する適応性を検証する．なお，文献 [2] の履歴統計手法における連続使用優先係数 P，履歴依存係数 K，および，DB移動複製法における P',K' の値としては，それぞれ別のシミュレーションでの評価によって得られた最適値を用いている．

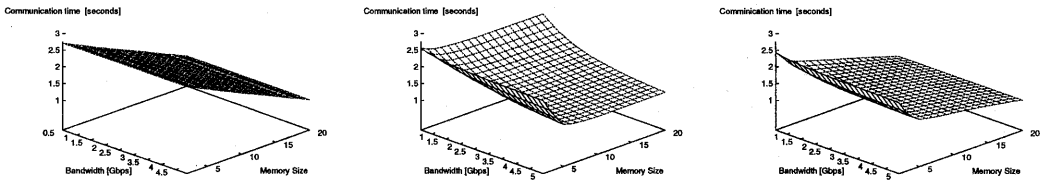
シミュレーションでは，各サイトが連続的に発生するトランザクションの回数を0回から3回の範囲でランダムに発生する．各トランザクションの操作対象となるデータベース数は1から5の範囲とする．操作対象となるデータベースは，連続使用を宣言されているデータベース以外はランダムに決定される．

3.2 シミュレーション結果

前節で示した環境に基づき，DB移動のための帯域幅と各サイトの主記憶領域のサイズを変化させて，各手法の平均通信所要時間を計算した．その結果を図2に示す．グラフでは，x軸がDB移動のための帯域幅，y軸が各サイトの主記憶領域のサイズ（データベースサイズの整数倍）をそれぞれ表している．この結果から，従来の静的な複製法は，ネットワークの帯域幅の影響は受けないが，主記憶領域のサイズに大きく影響を受けることがわかる．文献 [2] の履歴統計手法は，ネットワーク帯域幅によって性能が大きく変化する．一方，本稿で提案したDB移動複製法は，他の二手法ほどではないが，帯域幅と主記憶領域のサイズの両方の影響を受けている．

各手法の優劣を明確にするために，上記の結果に基づき，各ネットワーク帯域幅，主記憶領域のサイズにおいて最適（通信所要時間が最短）となる手法を調べた．その結果を図3に示す．この図では横軸がDB移動のための帯域幅，縦軸が各サイトの主記憶領域のサイズ，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，'Log' は文献 [2] の履歴統計手法，'Replica' はDB移動複製法をそれぞれ表している．この結果から，主記憶領域のサイズが4以上の範囲では，ネットワークの帯域幅が大きく主記憶領域のサイズが小さい場合に，文献 [2] の履歴統計手法が最も良い性能を示し，それ以外の場合ではDB移動複製法が最適となることがわかる．また，主記憶領域のサイズが5以下の範囲では，DB移動複製法が良い性能を示している．このような結果になる原因としては，次の三点が考えられる．

1. 複製を用いる手法では，主記憶領域のサイズが大きくなるとより多くの複製を保持できるため，トランザクション処理をほぼローカルに実行でき性能が向上する．



(a) 従来の複製法　(b) 履歴統計手法　(c) DB 移動複製法

図 2: 各手法の平均通信所用時間

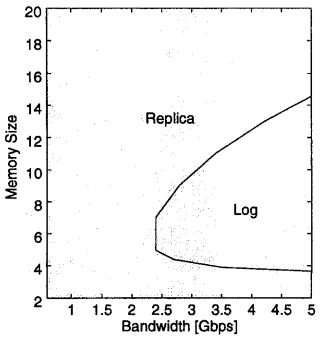


図 3: 最適手法の分布

2. ネットワークの帯域幅が大きくなると，DB 移動のための転送遅延よりもメッセージの通信回数の方が性能に大きく影響するようになるため，複製を用いる手法における複製間の一貫性保持のための 2 相コミットが性能を低下させる．
3. 主記憶領域がかなり小さい場合，履歴統計手法では，DB 移動を用いた処理を実行する際に領域確保のための DB 移動が頻繁に発生するため性能が低下する．

4 むすび

本稿では，DB 移動を用いて動的にデータベースの複製を再配置する手法を提案した．更に，シミュレーション評価によって，提案した手法の有効性を検証した．この結果より，提案した DB 移動複製法が，広範囲にわたって良い性能を示すことが確認できた．

今後は，提案した手法を実システム上で実装することにより，プロトコルオーバヘッドなどの実環境における様々な要因も考慮した評価を行う必要がある．

謝辞

本研究の一部は，文部省科学研究費奨励研究 (A)(09780380) 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] D. DeWitt, R. Katz, F. Olken, L. Shapiro, M. Stonebraker, and D. Wood, "Implementation techniques for main memory database systems," Proc. ACM SIGMOD'84, pp.1-8, June 1984.

[2] 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)，掲載予定.

[3] 原隆浩，春本要，塚本昌彦，西尾章治郎，"ATM ネットワークにおけるデータベース移動のためのデータベース位置管理手法," 信学論 (D-I)，Vol.J80-D-I, No.2, pp.137-145, Feb. 1997.

[4] 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分散データベースシステムにおける並行処理制御について," 情処研報，Vol.96, No.1, pp.179-186, Oct. 1996.

[5] 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.J79-D-I, No.4, pp.460-467, April 1996.

[6] M. Stonebraker, "Concurrency control and consistency in multiple copies of data in distributed INGRES," IEEE Transactions on Software Engineering, Vol.3, No.3, pp.188-194, May 1979.

[7] R.H. Thomas, "A mojority consensus approach to concurrency control for multiple copy databases," ACM Transaction on Database Systems, Vol.4, No.2, pp.180-209, June 1979.