1R－4

# 永続分散共有メモリ機能を提供するデータベースサーバ「わかし」のマルチメディア拡張 *

金子 邦彦，牧之内 顕文，金 泰勇
九州大学 大学院システム情報科学研究科 †

## 1 はじめに

本発表では，「わかし」のマルチメディア拡張におけるキーアイデアを報告する．わかしとは，我々が研究開発を行っているデータベースサーバのプロトタイプである[2]．わかしは，「分散共有メモリ」をベースとしているために，マルチメディアを扱うにはメッセージ数の削減が必要である．同時に２フェーズ・ロックを基礎とした従来のトランザクション管理方式の拡張も必要である．

## 2 わかしと分散共有メモリ

分散共有メモリとは、ネットワーク上の複数の計算機が１つのメモリ空間を共有したものだと定義できる．具体的には，ネットワーク上の各計算機に分散共有メモリの複製が作成され，互いの通信により各複製のメモリイメージの同一性が維持される（このことをメモリコヒーレンスという）．

メモリコヒーレンスが行われていることはアプリケーションから隠されているため，分散共有メモリへのアクセスは，普通のメモリアクセスと同様に行うことができる．このことは，分散環境におけるアプリケーションの開発をネットワークを意識せずに容易に行えることを意味する．従って，分散共有メモリの利用により，マルチメディアに限らず，分散アプリケーションを容易に開発できる[1]．

わかしにおけるメモリコヒーレンスは，「必要になったら通信を行う」という方針である．すなわち，最初，分散共有メモリの中身は空であり，アプリケーションからの要求に応じてデータ転送が行われ，アプリケーションのメモリ空間内にデータの複製が作成される．

わかし上のあるアプリケーションが分散共有メモリ内のデータ読み出しを始めて行う場合，内部的には図２のように２往復のメッセージ通信が行われる．メッセージの最初の１往復はロックに関するものであり，トランザクションの実現に必要である．次の１往復はデータ転送に関するものである．データ転送における通信先は，ロック要求の場合と違って，「最新の複製を持つサイト」となる．

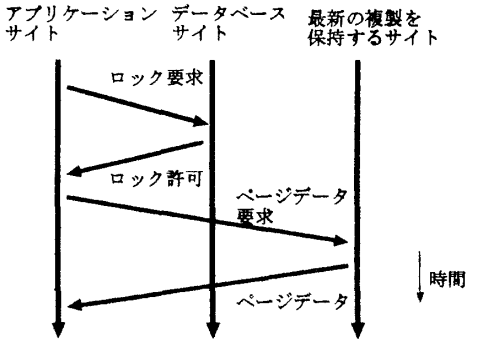


図 1: リード時におけるメッセージは２往復である

データ書き込みの場合には，複製が置かれているすべての計算機に対して「当該データは更新されたので，今の複製は無効である」ことを知らせるメッセージ (invalidation) が通知される．

## 3 バルクページ

わかしの分散共有メモリでは，ロック及び転送の単位は「データ」でなく，「ページ」である．すなわち，あるデータＡがアクセスされた場合，このデータＡを含むページ全体がロックされ，転送される．

わかしで，マルチメディアデータのような非常に大きなサイズのデータを扱うと，現状では，ページ単位でのメッセージ通信を繰り返す．ロック要求からページデータ到着までの間，アプリケーションの実行は中断されるため，大きいサイズのデータを効率よく扱えない．例えば，シーケンシャルリードでは，図２に示すように，ページの境界に達するごとにアプリケーションの実行が中断する．

大きなサイズのデータを効率よく扱うには，ロック及び転送をページ単位でなく，複数個のページをひとまとまりとした単位（バルクページ）で行うことが有効である．バルクページでは，メッセージサイズの総計はあまり変らないものの，メッセージ数は大きく減少する．バ

*Mechanisms for Multimedia Data in a persistent distributed shared memory server — Wakashi, Kunihiko KANEKO, Akifumi MAKINOUCHI, Taiyong JIN

†Graduate School of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-Ku, Fukuoka 812-81, Japan

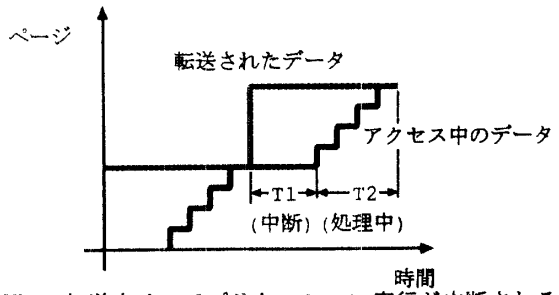

図 2: 転送中は，アプリケーション実行が中断される

ルクページは，シーケンシャルアクセスのように，アクセスするデータが連続するページ上にある場合に最も有効である．1バルクがNページであるとすると，シーケンシャルリードでは，メッセージ数が 1/N に減少する．

アプリケーションの待ち時間は，データサイズxに対する1次式 ax+b (aは帯域の逆数であり，bはレーテンシである) で近似できる．ATMのような最近のネットワークでは，レーテンシよりも帯域が大幅に向上する傾向にある．従って，メッセージ数を減少させることが，待ち時間の減少に有効なアプローチであるといえる．

バルクページを単純に実装すると，ロックの単位が大きくなり，結果としてトランザクションの並行度の減少及び性能の低下を招く．そこで，我々は次のような工夫を考えている．

1. ロックモードの拡張：ロックモードとして，リード，ライトの他に「予約」モードを導入する．実際にアクセスされたデータページのみにリード／ライトロックが行われ，同一バルク内の他のページには予約ロックのみが行なわれる．
2. ロックプロトコルの拡張：あるプロセスがすでに予約ロック済のページにアクセスする場合，「予約ロックをリード（またはライト）ロックに変更する」というメッセージがデータベースサイトに送られる．この間アプリケーションの実行は中断しない．
3. invalidation プロトコルの拡張：すでにあるアプリケーションPによって予約ロック済のページに，他のアプリケーションがライトを行う場合，「Pの予約ロックの取り消し」がPに通知される．

## 4 マルチメディア用トランザクション

わかしでのマルチメディア順再生では，図3に示すように，データを使用するプロセスと別に，データ要求用のプロセスを実行することで，単位時間あたりに表示できるデータが増える．転送にかかる時間と，表示にかかる時間の比を t1:t2 にように表すと（図3参照），全体の処理時間が max(t1,t2)/(t1+t2) だけ速くなる．

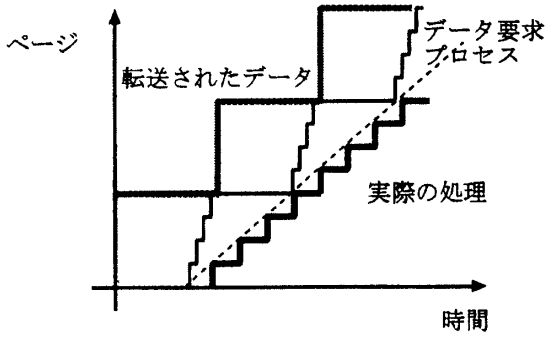

図3．マルチプロセスとして順再生を実現すると性能が向上する

CADなど，同一データを複数の人間が同時にリード／ライトしつつ共同作業を行うような状況においては，他の人間が作業中のデータへのアクセス (dirty read) が許されねばならない．従って，各個人の作業を個別のトランザクションとして実現することは，厳密な意味ではできない．

以上の2例のように，マルチメディアでは，複数のプロセスによる並列アクセスがあり，従来の2フェーズ・ロックを基礎としたトランザクション管理では対応できず，その拡張も必要である [1].

我々は，ネットワーク上に分散された並列入れ子トランザクションによる解決を考えている．具体的には，（1）一般に「1プロセス上の命令系列」と考えられてきたトランザクションを，ネットワーク上の複数プロセスに拡張，（2）個々のプロセスの命令系列（あるいはその1部分）は，入れ子トランザクションのサブトランザクションとして実現し，サブトランザクションのコミット時には計算結果が他のプロセスへ配送される．結果として，並列処理が実現される．

## 5 おわりに

マルチメディアでは，サイズが非常に大きく，ある意味での並列処理も必要であるから，本稿に示したバルクページ，並列入れ子トランザクションのアイデアが有効であると考えられる．

## 参考文献

[1] 金子邦彦，牧之内顕文，有次正義，“Multimedia Applications using a Database Programming Language - INADA”，1996 IEEE International Conference on Multimedia Computing and Systems, pp.458-461, 1996.

[2] G.Yu, K.Kaneko, G.Bai, A.Makinouchi, “Transaction Management for a Distributed Object Storage System WAKASHI - Design, Implementation and Performance,”12th Int'l Conf. on Data Engineering, 1996.