データベースシステム 104−10
（１９９５．７．１８）

マルチメディアデータベースサーバＩＮＡＤＡと その上のマルチメディアデータベースモデル

金子邦彦，進英二，牧之内顕文

九州大学工学部情報工学科
〒812-81 福岡市東区箱崎 6 − 10 − 1

現在，我々の研究室では，ネットワーク上での peer-peer タイプの分散マルチメディアデータベースサーバＩＮＡＤＡの研究開発を行なってきたところである．

本論文では、まず，ＩＮＡＤＡの概要と特徴を述べる．次に，ＩＮＡＤＡ上で蓄積ビデオ，ライブビデオ，立体画像などマルチメディアデータを扱うための課題について述べる．マルチメディアでは多数の基本的オブジェクトから構成される複合オブジェクトを扱う必要がある．そこで，工学分野向けの代表的なデータベースベンチマークである OO7 ベンチマークで評価を行う．最後に，ライブビデオ (テレビカメラとマイク) や黒板など随時連続的に更新が発生するメディアの取扱いについて考察する．

A Multimedia Database Server — INADA, and its Multimedia Database Model

Kunihiko Kaneko, Eiji Shin, and Akifumi Makinouchi

Department of Computer Science and Communication Engineering
Kyushu University
6 − 10 − 1 Hakozaki Higashi-Ku Fukuoka, 812-81 Japan

We are now designing and developing a multimedia database server for distributed environment via network, named INADA. INADA has C++ interface, and INADA is a platform for multimedia applications written with C++.

In this paper, we explain about the overview and feature of INADA. Then, we also summerize problems which you meet when you use INADA for multimedia such as stored-video, live-video and 3-D computer graphics data. Finally, we report the result of performance test of INADA using of the OO7 benchmark.

1 はじめに

近年、静止画，ドキュメント、音声、音(オーディオ)，蓄積ビデオ，ライブビデオ，2次元グラフィックス，立体画像，音声，文字などのマルチメディアを計算機上で扱うことに注目が集まっている．

単に，マルチメディア機器を計算機でコントロールするのではなく，マルチメディアデータをフルデジタル化して計算機で扱うことで，サーバマシン上のマルチメディア情報を複数の利用者に対して同時かつ即時に提供するというオンデマンド型の処理だけでなく，複数の場面から構成された番組を利用者の希望に応じて異なる順序で表示するような柔軟な処理や，質問，討論，意見，問い合わせなど双方向の通信を行うなどのより柔軟な形態を持ったアプリケーションの開発が容易となる．

これらマルチメディアを計算機上で扱うことが可能となった背景としては，個別のメディアを計算機上で可能とするための各種メディア入出力用の専用ハードウエアの普及が挙げられる．例えば，動画像のデータは巨大なのでそのままでは計算機が処理を効率良く行うことが難しい．近年，MPEG, MPEG2 などの動画像ディジタル符号化技術の進歩とそのハードウエア化が進み，テレビカメラやVTR から得られるアナログ動画像を実時間でデジタル化・圧縮・符号化することが可能となったことから，動画像を計算機で扱うことが容易となった．

各種メディア入出力用の専用ハードウエアの普及の一方で，ネットワークの高速化・大容量化，主記憶・二次記憶の大容量化も進んでいる．現在，代表的なローカルエリアネットワークであるイーサネットは，10Mbps から 100Mbps の容量を持つが，新しいネットワーク技術である ATM 網は，150Mbps の容量を実現し，将来的には数 Gbps が可能とされている．さらに，ATM 網によって構築された広域ネットワークは，インターネットのようなルータを用いて構成された広域ネットワークと比べて伝送の遅延時間がかなり少ない．

以上のようなマルチメディア / 情報スーパーハイウエイ時代の到来を踏まえ，我々は，ATM 網のような高速・大容量ネットワークで互いに接続された複数のワークステーション上で，蓄積ビデオ，ライブビデオ，立体画像を扱ういくつかの実験システムの構築・実験と，これらマルチメディアデータの効果的格納・配布・応用の研究を開始した．

数多くの計算機がネットワークで結合されることで，マルチメディアを同時に扱えるような統合されたマルチメディアシステムが一層重要になると予想される．従来の単一メディアの格納・配布・応用の技術を統合し，マルチメディアデータを統一して扱うにはデータベースシステムの概念が有効であることを従来，多くの研究者が指摘してきた [Ma95].

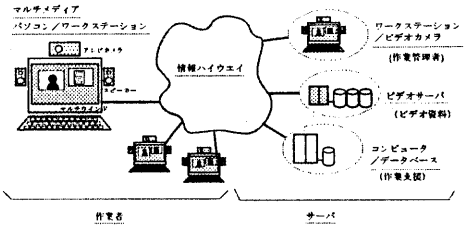


図 1: 遠隔作業システム

そこで，我々は，マルチメディアデータを扱うアプリケーションの基盤としてのマルチメディアデータベースサーバ INADA の研究・開発を開始した．ネットワークの個々の計算機のメモリ，二次記憶，プロセッサ計算機資源を利用してマルチメディアアプリケーションを効率よく実現できるような基盤として動作するよう INADA を開発することを目指している．

本発表では，INADA のストレージ管理用サブシステムである WAKASHI のプロトタイプの概要と特徴を述べる．マルチメディアでは多数の基本的オブジェクトから構成される複合オブジェクトを扱う必要があり，オブジェクト指向データベースシステム用の工学的分野における代表的なベンチマークである OO7 ベンチマークで評価を行う．最後に，ライブビデオ (テレビカメラとマイク) や黒板など随時連続的に更新が発生するメディアの取扱いについて簡単に考察する．

2 マルチメディアデータベースシステムの課題

マルチメディアアプリケーションにはいろいろな種類が考えられるが，我々は，将来的に試作したい実験システムとして “遠隔作業システム”，“マルチメディア 4 次元データベース”，“マルチメディア共在仮想空間システム” の 3 つを考えている．

2.1 遠隔作業システム

パソコン／ワークステーションにテレビカメラとマイクを接続する．さらに，相互に高速・大容量のネットワークで繋ぎ，高速なデータの検索・配送

を行えるようにする (図 1). この上で，次の 3 つの機能を持った遠隔作業システムの実現を目指す．(1) ＭＰＥＧやＭＰＥＧ２などの動画像デジタル符号化技術をベースとしたビデオ（＝動画像＋音声）を効果的に格納・検索・転送する．(2) ライブビデオ（テレビカメラとマイク) のデータを高速なネットワークを利用して実時間で転送する (3) 単にサーバ側から動画像，ドキュメントを配送するという一方向の通信だけでなく、適宜、討論，打ち合せ，資料の交換など双方向の通信を行う

2.2 マルチメディア 4 次元データベースシステム

“世界の事物の時空間としての存在，時空構造・属性 (モデル)” と “コード化された動画像と音声” とは別個の存在としてとらえることが可能である．そこで，本システムでは，動画像を単なる画像ではなく、実体とその動きに関するモデルを持った画像として考え，より意味論的検索・編集を可能とすることを目的とする (図 2)．例えば，” 野球のビデオ” に登場するチーム、選手、球場などの実体情報と，選手の動作 (ヒット、ホームラン、フライ、ゴロ、ダブルプレイ、バント、盗塁) を野球のビデオのモデルとしてデータベースに格納しておき，“野球中継の途中から見始めた場合に、決定的シーンを捜し出す “，“ある特定の選手の過去のプレイを再生できる” などの機能を実現する．

これらモデルのデータは，4 次元空間に存在する実体の属性としてとらえられる．そこで，時空属性を持った実体，およびそれらを被写体や音源とする画像や音声を格納するデータベースシステムをマルチメディア 4 次元データベースと呼び，これを研究開発する．実体のモデルは，外界の 3 次元的な空間構造をモデル化している．実体の動きのモデルは時間のモデルと関係がある．

立体画像 (コンピュータグラフィックス＋アニメーション) は，画像生成のための実体のモデルを持つ．一方，動画像のコードそのものは実体のモデルを持たない．動画像では，時間関係はシーケンスとして認識される．まず，これら時系列メディアを単なる時間順の列ではなく、カット，セグメントなどの意味のある区間に階層的に分割し，“並べ変え” などの編集を容易にする．動画像に関する実体とその動きのモデルは，動画像から画像処理で抽出され，データベースへの格納される．

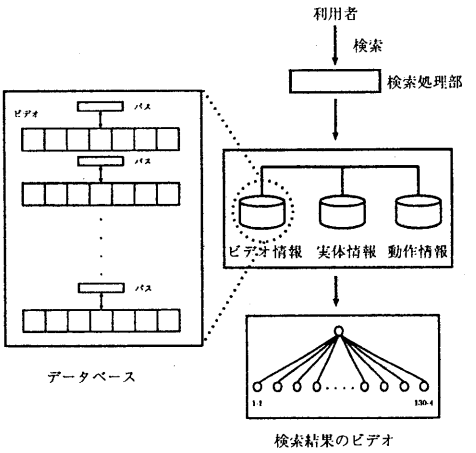


図 2: ビデオデータの検索処理の流れ

2.3 マルチメディア共在作業空間システム

遠隔地にいる複数のユーザ (作業者) によるネットワークを介して協同作業を行うための，本システムは，作業者が 3 次元の仮想会議室・作業空間に入り込み共に情報を共有・交換を可能とするための『共在作業空間』を実現する．

共在作業空間は，利用者が作業をするための 3 次元仮想空間である．共在作業空間は、複数の遠隔地ユーザによって共有され，あたかも同じ空間に同時に存在している感覚で共同作業を遂行する感覚を与える．共在している作業者は空間内の事物・マルチメディア情報を共有し，自由にアクセスすることが可能であり，作業者相互の交信・交流 (接触，会話，手による指示など) も可能である．

CSCW: (Computer Supported Cooperative Work) のように，ネットワークに結合されたコンピュータ端末を使って遠隔地にいるユーザが協調的に作業を行うシステムでは，データを互いに交換することによる協同作業である．互いに「同一の場所に存在して」の協同作業ではない．

共在作業空間の実現には，映像・音声を合成するための 4 次元実体のデータベースへの格納，実時間伝送，データベースへの仮想現実的インタフェースが必要である．

2.4 マルチメディアデータベース

以上のようにマルチメディアアプリケーションにはいろいろな種類がある．従って，マルチメディ

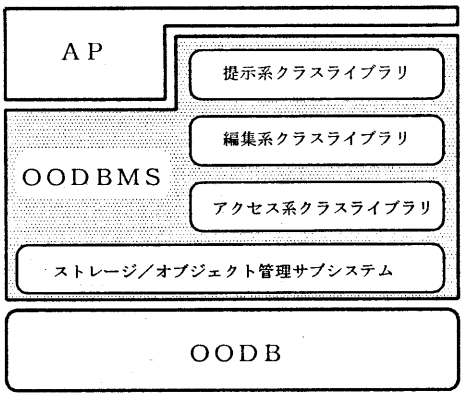


図 3: マルチメディアデータベースシステムの構成要素

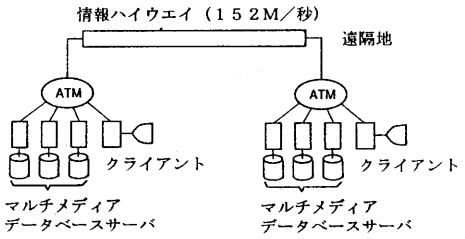


図 4: マルチメディアデータベースと高速・大容量ネットワーク

アアプリケーションを容易に開発するためのプラットホームとしてのデータベースシステムが重要となっている.

マルチメディアデータのような複雑な構造を持ったデータは，オブジェクト指向で自然に表現できることを多くの研究者 [Ma95] が指摘してきた．図 3 のように，オブジェクト指向データベース上にマルチメディア格納・検索・編集・表示用のクラスライブラリを整備しておくことで，アプリケーション部分の大きさは小さくなり，アプリケーションの開発は容易になる.

図 3 のストレージ／オブジェクト管理サブシステムには，マルチメディアデータを扱う場合に必要となる下記の基本的な機能を持たねばならない.

- 分散環境

  マルチメディアデータベースシステムは，図 4 のように分散環境で動作する

- データベースの分散

  各利用者固有のデータベースは，その利用者のサイトに格納することが望ましい．しかし，それらは統合されなくてはならない.

- 実時間性

  遠隔地とのビデオの交信を自然に行うには，ビデオのような長大データを高速かつ連続的に伝送する必要がある.

- データの共有

  複数のクライアントがマルチメディア情報を共有する.

- マルチメディア用クラスライブラリの実装の容易さ

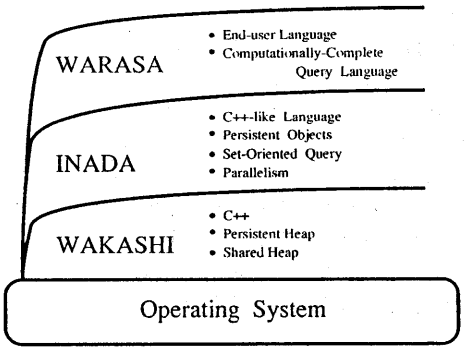


図 5: 出世魚を構成する 3 つのレイヤ

近年，WWW，xmosaic などを利用して，サーバマシンに蓄積されたマルチメディア情報を複数の利用者に対して同時かつ即時に提供するというオンデマンド型の処理を容易に実現できるようになってきた．WWW を利用して，ネットワーク上への情報発信も活発に行われるようになってきた．WWW ではマルチメディアを HTML フォーマットのファイルとして扱っているため，格納，検索，編集，提示 (プレゼンテーション) に関するメディア固有の処理は HTML ファイルフォーマットを意識して書かねばならない．従って，WWW 上のマルチメディアアプリケーションの開発は必ずしも容易ではない.

3 出世魚

マルチ CPU、高速ネットワーク、大容量主記憶など計算機環境の変化とともに、従来のオブジェクト指向データベースシステムの欠点もいくつか指摘されるようになってきた [BaMa94]．そこで，九州

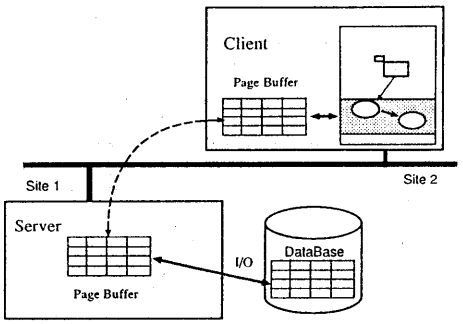


図 6: Page-Server (E) のストレージ管理方式

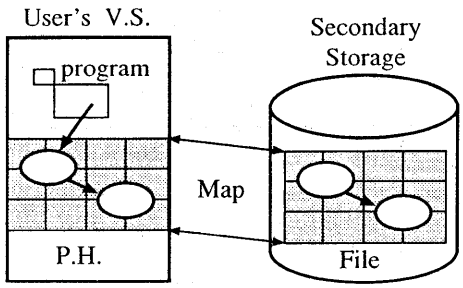
大学牧之内研究室では，オブジェクト指向永続プログラミング言語とその開発環境『出世魚』の研究・開発を進めてきた．出世魚は，図 5 のように WAKASHI，INADA，WARASA の 3 つのレイヤからなる階層構造を持っている．現在，我々の研究室では，『出世魚』の WAKASHI，INADA 層をベースとして，ネットワーク上でのマルチメディア・サーバＩＮＡＤＡの研究開発を開始した．

3.1 WAKASHI のストレージ管理方式

WAKASHI は，ネットワーク上に存在する多数のメモリ領域 (ヒープ) を管理するストレージ管理サブシステムである．

従来のオブジェクト指向データベースのストレージ管理方式は，おおまかに (1)Page-Server，(2)Object-Server(O_2 など)，(3)File-Server の 3 種類に分類することができる．いずれの方式も，応用プログラム (クライアント) が，サーバへデータ要求を実行すると，要求されたデータがサーバからクライアントに転送されるという方式である．多くの場合，データ転送は図 6 のようにバッファを介して行われる (バッファ方式)．バッファ方式では，サーバはデータベース内のデータをバッファへ読み込み，データをクライアントのプログラム領域の変数領域に転送する．主記憶上の変数領域のデータ形式と二次記憶上のデータ形式は一致しないので，変換が必要となり，その分効率が低下する．

最近の OS では，二次記憶と主記憶間の転送方式として，図 7 のように，仮想メモリ管理機構を利用して二次記憶上のデータベースファイルを仮想記憶領域に写像する方式 (仮想メモリ方式) が実現されるようになってきた．



V.S.: Virtual address Space

P.H.: Persistent Heap

図 7: 仮想メモリ方式

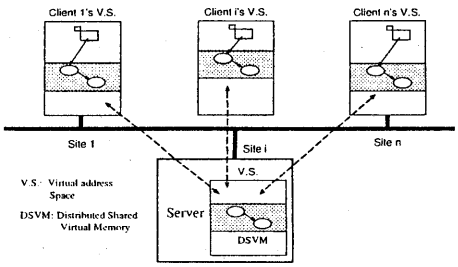


図 8: 分散共有メモリ方式

一方，ネットワークで相互接続されたプロセッサ群において分散されたデータの共有を効率良く行うことができる方式として，図 8 のように，各プロセッサのローカルメモリをまとめて仮想的に 1 つの共有メモリが存在するかのように見せる機能 (分散共有メモリ) [NiLo91] が提案されてきた．

WAKASHI では，図 9 に示したような，仮想メモリ方式と分散共有メモリを組み合わせることで，データベースファイルをクライアントの存在する遠隔サイトの仮想空間へ写像する方式 (分散仮想共有メモリ方式) を実現している [BaMa94]．仮想メモリ方式を用いたので，主記憶上のデータ形式と二次記憶上のデータ形式がほぼ一致し，永続データをバッファ方式よりも効率良く扱うことができる．

各サイト上のクライアント（ユーザプログラム）は共有メモリ（グローバルヒープとして見える）上に情報を書き、またその上の情報を読む（プログラム的には、通常のプログラムデータの読み書きと全く同じ）ことが可能である．クライアント側のプログラムは，他のクライアントプログラムとマルチ

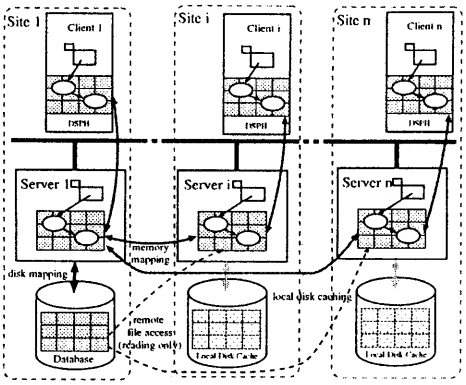


図 9: WAKASHI のストレージ管理方式

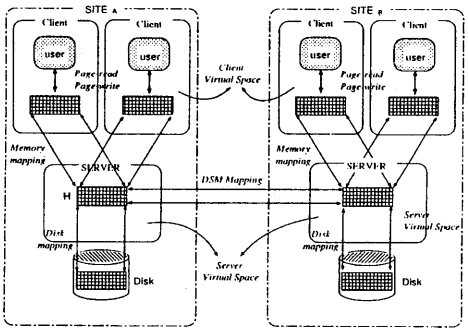


図 10: WAKASHI のアーキテクチャ

メディアデータを仮想空間上に共有でき，ローカルなデータベースと遠隔のデータベースをネットワークを意識せずに操作できる．

WAKASHI のグローバルヒープは，ページ単位に分割されて管理されている．ロック，転送はこのページ単位で行われる．ユーザは，このグローバルヒープ上に C++ オブジェクトを作成することが可能である．しかも，トランザクション管理，ロック，ヒープの転送，メモリコヒーレンス管理などの管理は WAKASHI サーバが自動的に行い，アプリケーションは明示的にロック命令を呼び出す必要はない (Paged-Object サーバアーキテクチャ)．

3.2 WAKASHI のアーキテクチャ

従来のクライアント / サーバアーキテクチャに基づくオブジェクト指向データベースの多くは集中型クライアント / サーバアーキテクチャのため，サーバに処理が集中し，クライアントマシンのクライアントマシンの主記憶・二次記憶など，ネットワーク上の資源を十分には活用できない．ネットワークの高速化とともに，サーバのボトルネックが問題になると予想される．

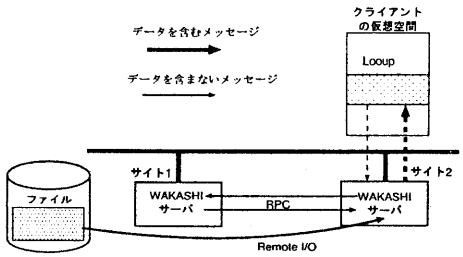


図 11: WAKASHI による遠隔 lookup

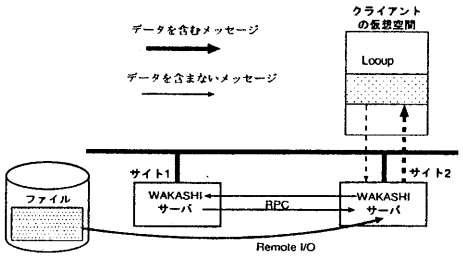


図 12: OO7 ベンチマーク用データベースの構造

WAKASHI では，図 10 のように，各分散サイトに、メモリを管理し、情報を共有するための WAKASHI サーバが載る．各サーバは全く対等であり、いわゆる peer-peer タイプのサーバシステムである．サーバマシンを固定せず，同一のヒープにかかわる処理をそのヒープを操作する全てのマシンにほぼ対等に分散する．

WAKASHI では，一度遠隔サイトへデータが転送されると，そのデータに関する処理の多くは遠隔サイトで実行されるようになる (図 11)．WAKASHI サーバのヒープマネージャは各ヒープ毎に独立のスレッドが並列実行するマルチスレッド方式である．

4 OO7 ベンチマーク

我々は，WAKASHI を，オブジェクト指向データベースシステム用の代表的なベンチマークである OO7 ベンチマーク [CDN93] を用いて評価した．OO7 ベンチマークは，CAD などの工学分野におけるデータベースの性能評価のためのベンチマークである．007 ベンチマークは，互いに関連のある数万から数十万個の部品をデータベース上に表現し，Lookup, Traverse, Insert, Update という 4 種類

の操作を行いトランザクションの開始から終了までの経過時間を測定する．

マルチメディアでは，基本的実体オブジェクトが多数組合わさった複合オブジェクトとして表現されることが多い．007 ベンチマークでは，(1) 数多くの部品から構成される複合オブジェクトに対して性能測定を行う，(2) Text データとしてある程度サイズを持ったオブジェクトを扱う，という 2 つの理由から，マルチメディアデータベース用の性能評価の第 1 歩としてふさわしい．

Machine Parameters	
mode	Sparc 20
number_of_cpus	4 per site
page_size	4 Kbytes
swap_space_size	180 Mbytes per site
disk_size	1000 Mbytes per disk
number_of_disks	1 per site
disk_random_access_time	20 ms
network_speed	10 Mbytes per sec.
System Parameters	
OperatingSystem	SunOS 2.3
NetworkProtocol	TCP/IP
Database Parameters	
heap_size	100 Mbytes
number_of_heaps	3

表 1: 007 ベンチマークのパラメータ

今回の実験では，表 1 の条件で測定を行った．007 ベンチマークのプログラムは，ウイスコンシン大学より入手し，INADA で動作するように変更を加えた．

今回の測定では，007 ベンチマークのうち，Traverse #1 と Traverse #2 の測定を行った．今回，コールドスタート，ホットスタートの 2 条件で測定した．コールドスタートは，データベースがまだメモリに読み込まれていない状態である．ホットスタートは，同一のベンチマーク操作を何度か繰り返し，データの多くが二次記憶からメモリ上へ読み込まれた状態で再びベンチマーク操作を実行した場合である．遠隔サイトのデータベースをアクセスする場合，ホットスタートでは，WAKASHI は，データの多くを遠隔サイトの二次記憶から自サイトのメモリ上へ転送する．

図 13 には，Traverse #1 の実行結果，図 14 には，Traverse #2C の実行結果を示している．

測定結果から，コールドスタートでは，ローカルへのアクセスと，リモートへのアクセスでは，3 倍の差になっていることが分かる．これは，データ転送と，転送されたデータを仮想空間へ書き込む処理のオーバーヘッドである．

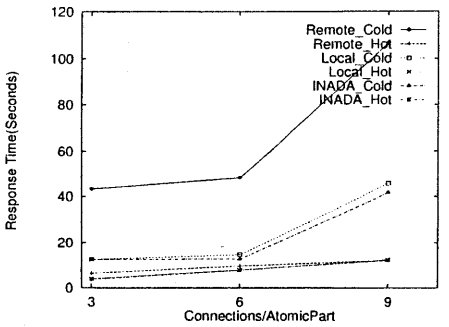


図 13: Traverse #1

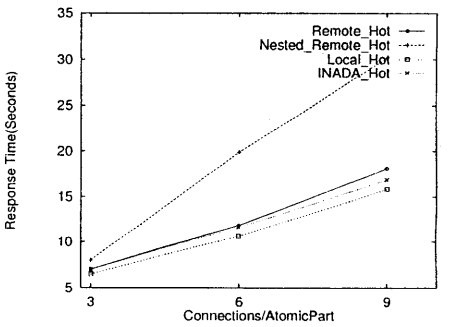


図 14: Traverse #2C

当然のことながら，ホットスタートは，コールドスタートよりも高速になっている．これは，ホットスタートではすでにデータが二次記憶から主記憶へ読み込まれた状態になっているからである．リモートとホットスタートの組み合わせでは，ローカルとホットスタートに匹敵する性能が得られた．

最後に，システムコール mmap() で確保されるヒープと，WAKASHI のヒープで測定し，比較したところ，数%の差しかないことが分かった．従って，WAKASHI のトランザクション管理のオーバーヘッドは小さいことが分かる．

5 課題

テレビ会議システムでは，ビデオ，黒板などのマルチメディア情報を遅滞なく他のサイトに伝えねばならない．

マルチメディアデータの配送従来の並行処理制御（2PLなど）では、書き込みロックのかかったページを、他の利用者は読み出すことができない従って，ライブビデオ、黒板など同じデータを複数のユーザが共有する局面では、書き込み中のデータを他の利用者は読み出すことができない．

そこで，我々は，マルチメディアトランザクションの導入を考えている．まず，マルチメディア用に(1) Write-Only トランザクション TW，(2) Read-Only トランザクション TR を導入する．そして，従来のトランザクションの4条件 Atomicity, Consistency, Independency, Durability を緩和し，TW では，A, C, D の3条件のみ，TR では，A, C, D の3条件のみで，さらに C では書き込み途中のデータを読み出し可能とするというように緩和する．

グローバルヒープのうち更新されたページは，自動的に他のサイトへ配送される必要がある．それには，WAKASHI サーバが一度ロックを開放したページを再びアクセスしたとき，そのページが更新されているかどうかの確認を行う必要がある．

6 おわりに

本発表では、蓄積ビデオ，ライブビデオ，立体画像などマルチメディアデータを扱うための基盤としてのマルチメディアデータベースサーバＩＮＡＤＡのストレージ管理用サブシステムである WAKASHI のプロトタイプの概要と特徴を述べ，分散共有仮想メモリ，オブジェクトページサーバ，peer-peer アーキテクチャという3つの WAKASHI の中心的な技術についてその利点を述べた．

次に，OO7 ベンチマークによる WAKASHI の評価では，遠隔サーバとローカルサーバを (ネットワークの伝送時間を除いては) ほぼ性能上同じように利用できることが分かった．

最後に，最後に，ライブビデオ (テレビカメラとマイク) や黒板など随時連続的に更新が発生するメディアの取扱いについて，マルチメディア用トランザクションの考えを簡単に述べた．

7 謝辞

WAKASHI の設計，実装に大きく協力いただいた富士通九州通信システム石井 孝明氏，岩崎 孝司氏，酒井 重徳氏，富士通研究所の白 光一氏に感謝する．WAKASHI のデバッグと OO7 ベンチマークの実装，測定に大きく貢献いただいた九州大学牧之内研究室の Yu Ge 氏，山本 完氏に感謝する．

参考文献

[BaMa94] Bai,G., and Makinouchi,A., "WAKASHI/D: A Distributed Paged-Object Server for Storage Management of New Generation Databases," Proc. of the Int'l Symposium on ADTI, Nara, 1994, pp.137-144.

[CDN93] Carey,N.J.,DeWitt,D.,J., and Naughton,J.F., "The OO7 Benchmark," Proc. of SIGMOD, 1993, pp.12-21.

[Deu90] Deux,O., et al. "The Story of O2," IEEE Trans. on Knowledge and Data Engineering, Vol.2,No.1, March 1990,pp.91-108.

[INADA94] INADA User Guide, Release 1.0, Dept. of Computer Science, Kyushu University, 1994.

[LL91] Lamb,C., Landis,G., Orenstein,J., Weinreb,D.J., "The ObjectStore Database System," Communications of ACM, Vol.34, No.10, October 1991.

[Ma95] Yoshifumi Masunaga, "A Temporal Expansion to the Multimedia Object Model in OMEGA," Proc. of the 4th Int. Conf. on Database Systems for Advanced Applications, pp.430-440, 1995.

[NiLo91] Nitzberg, B., and Lo,V., "Distributed Shared Memory: A Survey of Issues and Algorithms," IEEE Computer, August 1991,pp.52-60.

[YBGM95] Yu,G., Bai,G., Gao,J., Makinouchi,A., "Database Recovery for Transaction Management in WAKASHI," Proc. of the 50th IPS Conference of Japan, March 1995.