オブジェクトデータベース「出世魚」のODMG対応*

4R－1

山本 完$^{\phi}$，金子邦彦$^{\phi}$，有次正義$^{\varphi}$，牧之内顕文$^{\phi}$

$^{\phi}$ 九州大学大学院システム情報科学研究科知能システム学専攻

$^{\varphi}$ 群馬大学工学部情報工学科†

概要

複雑なデータをワークステーションクラスタ上で分散並列に処理できる分散並列オブジェクトデータベースの重要性が高まりつつある．我々の研究グループでは，分散並列オブジェクトデータベース「出世魚」を研究している．

一方，近年，オブジェクトデータベースの標準化 (ODMG標準) が行なわれつつある．我々のシステムもこの標準に従う予定であるが，この標準には分散並列機能については対応がない．

今回，ODMG標準に沿いつつ，分散並列機能を利用可能な機構を実装したので，その報告を行う．

1 ODMG標準

これまでのオブジェクトデータベースは，それぞれにプログラミングインターフェースが異なっており，アプリケーションの移植性，相互運用性は低かった．

こうした状況を改善すべく，ODMG(Object Database Management Group) 標準 [1] が提案された．オブジェクトモデル，オブジェクト定義言語 (ODL)，オブジェクト問い合わせ言語 (OQL)，特定言語 (C++, Smalltalk, etc) のためのバインディング (オブジェクト操作言語 (OML)) の標準が定められている．

1.1 C++バインディング

C++を元にしてデータベース内のオブジェクトを扱うための操作構文を追加したオブジェクト操作言語．

1. データベースとクラスタ
   ODMG標準は，アプリケーションから見たインターフェースを統一しようという規格であるため，内部の実装についてはほとんど言及されてない．データベース内のクラスタについては，唯一，ユーザが新しいオブジェクトを既に生成済みのオブジェクトの「近く」に生成するよう指定する構文があるだけである．「近く」は，実装定義とされている．
2. 一時オブジェクトと永続オブジェクト
   一時オブジェクトは，C++言語が提供する変数（オブジェクト）である．これに対し永続オブジェクトはデータベースシステムが提供し，その生存期間がそのオブジェクトが生成されたプロセスを越えるものであり複数のプロセスにより共有される，とされている．

2 出世魚

出世魚は，階層化されたシステムで，分散ストレージサーバ WAKASHI とその上の C++ を拡張した永続プログラミング言語 INADA からなる．

C++ におけるヒープ領域に加え，INADA では2種類の分散共有可能な GH(Global Heap) 領域を提供する．1つは永続化可能な GH 領域 PGH(PersistentGH) で，もう1つは揮発な GH 領域 VGH(Volatile GH) である．これらの GH 領域は WAKASHI の分散共有メモリの機構を利用し，実現している．GH 領域のイメージを図1に示す．このGHを用い「出世魚」では，

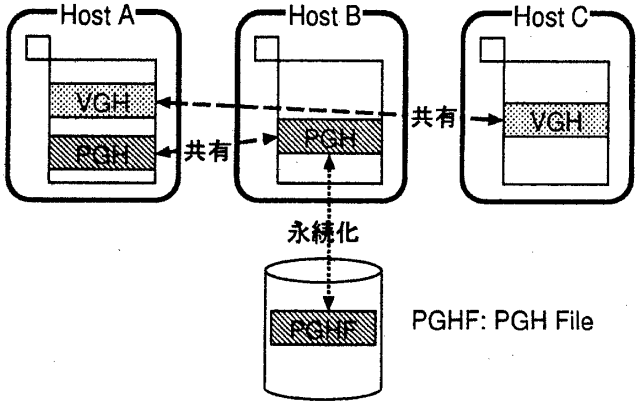

図 1: GH 領域の共有と永続化

(1) 遠隔のオブジェクトをネットワークを意識せずに扱える（分散透過），また (2) データベースを複数のパー

*Integrating ODMG interface into the object database "SHUSSEUO"

†Kan YAMAMOTO$^{\phi}$, Kunihiko KANEKO$^{\phi}$, Masayoshi ARITSUGI$^{\varphi}$, Akifumi MAKINOUCHI$^{\phi}$ $^{\phi}$ Graduate School of Information Science and Electrical Engineering,Department of Intelligent Systems, Kyushu University $^{\varphi}$ Department of Computer Science, Gunma University

ティションに分割し，各々のパーティションを分離させ（データベース分割）並列処理可能である．

3 C++ バインディングの実装

現在のODMG標準は，分散環境に対応してない．我々はODMG標準に可能な限り従いつつ（従ってODMG標準準拠のアプリケーションは運用可能）実装にいくつかの工夫を行うことで，分散透過性とデータベース分割の実現を行った．

3.1 データベースとクラスタ

ODMG標準では，オブジェクトを近接領域（クラスタ）に格納するためのインターフェースが規定されている．一般にクラスタは，ディスクの近接領域として実現されることが多い．が，我々は，データベースを"ヒープの集まり"として，クラスタを"ヒープ"として実現した(図2)．以下に詳細を説明する．

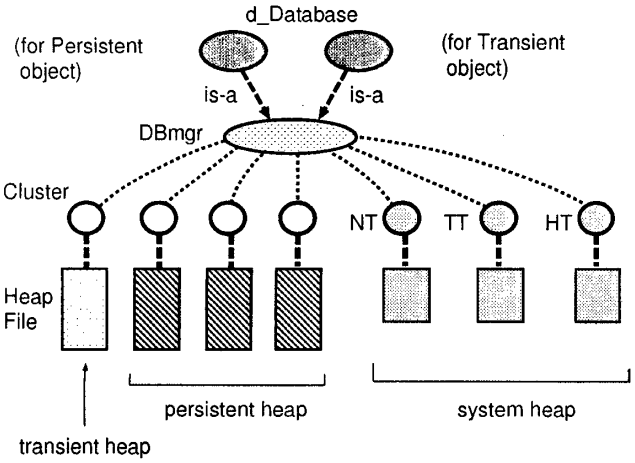

図 2: データベースの構造

1. d_Database
   データベースを管理するクラス．形式上，永続オブジェクト管理用と一時オブジェクト管理用の2つがある．

2. DBmgr(Database manager)
   ヒープ群を管理するためのオブジェクト．NT, TT, HTは，システムの使うヒープであり，その他はユーザから扱えるグローバルヒープである．

3. NT(Name Table)
   オブジェクトの名前を格納する

4. TT(Type Table)
   GH領域に格納されるオブジェクトの型名を保持する．

5. HT(Heap Table)
   ヒープのマップに必要なファイル名などの情報を保持する．

6. Cluster
   GH領域を管理するクラス．GH領域は，オブジェクトが格納される領域である．GH領域を仮想空間の任意の位置にマッピングできるように，ORT(オブジェクト参照テーブル)を使ってオブジェクトの間接参照を行っている．(図3)

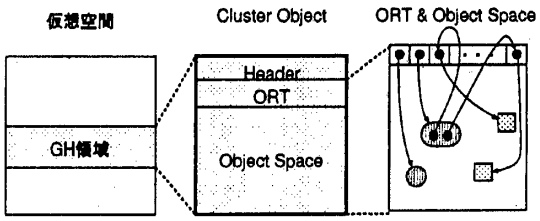

図 3: GH 領域

3.2 一時オブジェクト

ODMG標準の一時オブジェクトは，プロセスに固有でプロセスを越えては生存しないが，INADAの一時オブジェクトはVGH上に作られるオブジェクトであるため，複数のプロセスで共有され，複数のプロセスに跨って生存することができる．INADAは分散環境上での並行・並列プログラミングが特徴なので，一時オブジェクトの定義をODMG標準とは変えざるを得ない．

4 おわりに

「出世魚」の分散並列機能を生かしたODMG標準C++バインディングの実装法について述べた．今後の課題として，以下のものがあげられる．

- メタデータ
  ODMG標準で規定されている機能を実現するためには，現在のタイプテーブルでは不十分であるため，新たなメタデータの実現方法を検討する．
- ODL(オブジェクト定義言語)プリプロセッサ
  メタデータの抽出を行うため必要である．
- OQL(オブジェクト問い合わせ言語)
  容易なデータベースアクセスを提供する．

参考文献

[1] R.G.G.Cattell, "The Object Database Standard: ODMG-93 Release 1.2", Morgan Kaufmann Publishers, Inc.(1996)