

メディア変換機構を提供する分散永続オブジェクトシステムの設計

1H-02

高野了成†, 佐藤元信†, 早川栄一‡, 高橋延匡‡
†東京農工大学工学部 ‡拓殖大学工学部

1. はじめに

近年、計算機を文書作成や、研究開発などの発想活動支援に利用する機会が増加している。さらに、複数人での協調作業を支援するグループウェアなどの研究も行われている。このような計算機環境において、不定型なアイデアを逃さず記録し、必要な形式に加工し、データ間の相互関係を管理できることが重要である。

筆者らは、さまざまなデバイスやフォーマットのデータを格納、編集するためのメタアーキテクチャとして、メディア変換機構を特徴とする永続オブジェクトシステムを開発している [1]。本稿では、本方式の分散環境への対応について述べ、OS/omicon 第 4 版 (以下、V4) 上への実装方式について述べる。

2. 概要

2.1 作業モデル

不定型なアイデアやデータを逃さず捕まえ、後で加工できるように保存しておくことが重要である。そこで、capture-arrange-publish [2] の 3 フェーズから成る作業モデルを採用した。

(1) capture

さまざまなフォーマットのデータや、データ取得に伴う時間情報、位置情報など任意のメタデータを単一のデータエントリに格納できるようにする。

(2) arrange

スクリプト言語などを用い、メタデータを利用した検索やフィルタリング、データの加工を行う。また、データ間のリンクを容易に扱えるようにする。

(3) publish

WWW や紙への印刷物 (出力) として、または、他のモジュールへデータを出力する。

例えば、HTML 文書は閲覧することは容易だが、リンクが変更に対して静的である点や、入力方式がフォームに限られ、手書きストロークなどの多様な情報が扱えない点など、編集に適したフォーマットではない。そこで、各フェーズで、適切な操作が行えるように、データに対するメディア変換を実行できるようにする。

2.2 プラットフォームへの要求

上記のような作業を行うためのプラットフォームに求められる要求を次に示す。

- (1) データの共有が容易な分散永続オブジェクト機構
- (2) 拡張可能なメディア変換機構
- (3) データ操作用のオブジェクト指向スクリプト言語

(1) に関しては、本稿の以下の章で述べる。(2) に関しては、「意紙」サーバにおける属性変換機構 [1] を利用することで実現できる。

3. 設計方針

3.1 アドレス空間モデル

V4 は、セグメント ID (32bit)、オフセット (32bit) から構成される単一 64bit 2 次元アドレス空間を採用しており、筆者らは V4 上で永続オブジェクト機構を実現した [2]。永続オブジェクトはシステムで一意的なセグメント ID を持つ。この機構を利用し、セグメント ID の一意性を分散環境でも保証することで、図 1 に示すような単一 2 次元アドレス空間を持つ分散共有メモリを実現することができる。リモートマシンの永続オブジェクトに対してアクセスしたい場合は、永続オブジェクトの所有者を検索し、ローカルアドレス空間にコピーして、アクセスする。なお、永続オブジェクトを持つマシンをオブジェクトの所有者と呼ぶ。

3.2 プログラミングモデル

前述したように、本システムでは共有、操作の単位は永続オブジェクトである。永続オブジェクトは言語上からは変数として操作できる。V4 では、識別子はダイナミックリンク時に、変数の属性、識別子名、実際のアドレスなどの情報を含むリンケージテーブル経由でアドレスに変換される。分散オブジェクトに対しても、この機構を利用し、大域変数のバインディングを遅延評価することで、オブジェクトの所有者が不明でも、ランタイムが検索して、アクセス可能な状態にできる。したがって、アプリケーションは、オブジェクトを位置透過に扱えるようになる。また、オブジェクトの所有者が変わっても、プログラムを変更する必要はない。

しかし、すべての大域変数を分散共有するわけではないので、リンケージテーブルの属性として共有属性を追加する。これで、リンケージフォールト時に共有するのかどうか指定できる。

Design of a Distributed Persistent Object System with Media Converting Mechanism
Ryousei TAKANO †, Motonobu SATO †, Eiichi HAYAKAWA ‡, and Nobumasa TAKAHASHI ‡
† Tokyo University of Agriculture and Technology
‡ Takushoku University

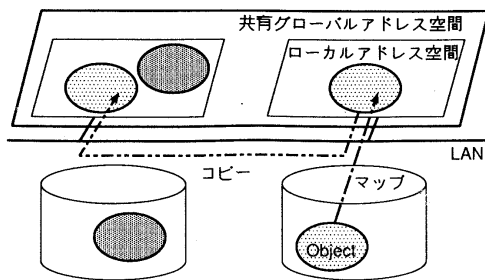


図1 アドレス空間モデル

4. 設計

4.1 システム構成

ハードウェア構成は、Ethernet で接続された PC/AT 互換機を想定している。次にソフトウェア構成を示す。

(1) 永続オブジェクトサーバ

ローカルマシン内における、永続オブジェクトとアドレス空間の対応、永続オブジェクトに対するメタデータを管理する。

(2) メモリドライバ

MMU 機構を利用して、パッキングストア操作を実装する。実装方法によって、ローカルマシンのディスク、またはネットワーク経由でリモートマシンからオブジェクトを読み込むことを選択できる。

(3) 分散オブジェクトサーバ

永続オブジェクトがどのマシン上に存在するかを管理する。また、同一マシン上に存在する永続オブジェクトに対するロックも管理する。

4.2 分散オブジェクトの管理

ローカル環境と異なり、分散環境で永続オブジェクトを共有する場合は、転送粒度、一貫性プロトコルについて考慮する必要がある。それぞれの設計について次に述べる。

4.2.1 転送粒度

転送粒度には、ページ単位とセグメント単位の二つの方式が考えられるが、それぞれの処理は、メモリドライバとして実現できる。また、メモリドライバは、セグメントごとに指定することが可能であり、セグメント作成時に転送粒度を設定することが可能である。

4.2.2 一貫性プロトコル

一貫性プロトコルには、一貫性の厳密さ、プログラミングの容易さ、性能などが異なるさまざまな方式が提案されており、アプリケーションの性質に合わせて選択できることが望ましい。

通常の単一アドレス空間を共有するページベース分散共有メモリでは、厳密な一貫性管理が必要であり、性能を出すことができない。一方、本方式では、Swizzling によるポインタ変換を行っており、リモートマシンにオブジェクトのコピーを作成することができるので、柔軟な一貫性制御が可能である。

今回は、基本的なリリース一貫性 (eager release consistency) [3] を採用することにした。

4.2.3 アクセス処理の流れ

共有オブジェクトに対するアクセス処理の流れを次に示す。

(1) 共有オブジェクトにアクセスする場合は、タスクが明示的にロックを獲得する (read only の場合は、例外である)。

(2) 共有オブジェクトに対するアクセスが発生した場合、セグメンテーションフォールトが発生し、分散オブジェクトサーバが共有オブジェクトの所有マシンを特定する。

(3) リモートマシンからオブジェクトがキャッシュされる。タスクはキャッシュに対してアクセスする。

(4) ロックの解放時に、ダーティページを書き戻す。

5. 関連研究

Dynamic Proxy Server [4] は、Proxylet と呼ばれるプログラムをネットワーク経由で追加することによって、動的な機能拡張を可能にしたプロキシサーバである。個々に機能が完結した Proxylet の組合せによって音声ファイルをストリーミングデータに変換するなどのメディア変換が可能になる。これに対して、本方式は WWW に特化しない汎用的であり、オブジェクトを共有アドレス空間に配置することで、拡張コード間における共有オブジェクトに対する協調操作が容易である。

6. おわりに

本稿では、メディア変換機構を提供する分散永続オブジェクトシステムの設計について述べ、OS/omicon 第 4 版における実装方法における議論を行った。今後は、本方式の実装、評価を行う予定である。

参考文献

- [1] 高野了成, 佐藤元信, 早川栄一, 並木美太郎, 高橋延匡: 多態的表現を可能にする永続オブジェクト管理機構, 情報処理学会研究会報告, 99-OS-81, pp.1-6, 1999.
- [2] Eiichi HAYAKAWA, Tomoyo SATO, Ryousei TAKANO, Motonobu SATO, Nobumasa TAKAHASHI, Flexible, Modular System Architecture for Supporting Creative Work, In Adjunct Conference Proceedings of HCI International'99, pp.163-164, 1999.
- [3] Andrew S. Tanenbaum, Distributed Operating System, Prentice Hall, 1995.
- [4] Michael Fry and Atanu Ghosh, Application Level Active Networking, In Proceedings of HIPPARCH'98, June 1998.