分散処理環境におけるオブジェクト実現方式

1M−6　谷林 陽一　佐藤 文明　中川路 哲男　水野 忠則

三菱電機(株) 情報電子研究所

1 はじめに

多数の計算機がネットワークで接続された分散環境では、多くの資源およびサービスを提供するサーバが物理的に分散して存在している。このような環境では、資源やサービスが物理的に分散しているため、集中型の環境に比べ、効率良く、信頼性の高いサービスを提供する可能性を持っている。しかしながら、大規模で複雑なシステムにおいてこの利点を発揮させるためには、システム内の資源を統一的に管理し、要求を最適なサービスに割り当てる機構が必要となる。

我々は、[1]で処理分散と資源管理を統合した分散処理環境の計算モデルを提案し、[2]でその実現機構についてインフラストラクチャとオブジェクトの関係を中心に検討した。本稿では、この環境で実際に動作するオブジェクトの実現方式について提案する。

ここで提案する方式は、オブジェクトを手続きとデータに分離し、システム内の複数のノードに分散配置する。これにより、要求を発するクライアントに対して、最適なサービスを提供することが可能となり、システム構成の変化に柔軟に対応できるシステムの構築が可能となる。

2 分散環境におけるオブジェクト実現

我々が分散環境を構築する際の目標は、

・位置透過性、
・スケーラビリティ、
・柔軟性

である。

分散システムは、様々な機能を持ったオブジェクトが協調動作しながら、利用者に様々なサービスを提供していると考えられる。オブジェクトは、データ領域とそれを操作する手続き、そしてその手続きを実行するものをカプセル化したものである。この考え方はプロセッサやメモリ、ディスクなどの資源が分散している環境でソフトウェアを構築するのに適している。

オブジェクトを分散環境で実現する場合、複数のノードに分散したオブジェクトを対応するサーバプロセスとして実現する方法が考えられる。これは、プロセス自身がモジュール、データ、実行の単位としてオブジェクトの性質を備えているからである。しかし、各サーバプロセスをオブジェクトとして扱った場合、以下のような問題がある。

実際の大規模な分散システムでは、スケーラビリティを確保し、信頼性や効率を向上させるために、オブジェクト(サーバプロセス)の複製を作り、ボトルネックを生じさせないようにすることが必要となる。その際、システム内に複製され分散した複数のオブジェクトを利用者には透過的に一つのオブジェクトとして見せる機構が別に必要となる。この透過性は、ノードの追加などによりシステムの構成が変化した場合にも保持されなければならない。さらに、膨大なデータを持つオブジェクトを複製する場合、複製先にもそのデータを全て格納する容量がなければならず、柔軟な複製ができなくなる。

3 オブジェクトの物理的な構造

我々は、以上の問題を解決するために、手続きとデータを分離し、これらを複数のノードに分散し、そろぞれが複製を持つことができるようにした。

3.1 オブジェクトの構造

ここでのオブジェクトは、利用者(クライアント)には通常のオブジェクト指向におけるオブジェクトと同じく、データ、手続き、手続きを処理する実行者が一つにカプセル化した論理的に一つのオブジェクトとして見える。しかし、実際のオブジェクトの内部はメソッド部(Method)とデータ部(Data)に分離され、物理的に分散配置されている。それぞれが複製を持つことも可能である。このようなオブジェクトとそれを管理する環境を図1に図示する。

・データ部

オブジェクトのデータ部分が蓄積されている。複製を持つことにより、効率および信頼性を向上させることができる。また、データ全体が膨大な量で一つのノードでは保持しきれない場合には、複数のデータ部に分割することができる。

・メソッド部

クライアントがデータを利用する場合は、必ずメソッドを介してアクセスしなければならない。メソッド部もデータ部と同様複数のノードに分散している。このため、ノードの負荷やデータ部との関係などに応じてメソッドを動的に選択することにより、クライアントに最適なサービスを提供することができる。

各オブジェクトのメソッド部およびデータ部は、複製も含め、互いの情報を保持しており、関係を自律的に管理している。このようにメソッド部とデータ部に分離することにより、例えば、大容量の記憶領域を持つノードにデータ部を高速なCPUを持つノードにメソッド部を配置したり、頻繁にアクセスを行なう利用者の近くにメソッドとデータの一部を配置するなど、利用形態に応じて最適なオブジェクト配置が可能となる。

3.2 オブジェクトの管理

複数のノードに分散配置されているオブジェクトのメソッド部およびデータ部は、インフラストラクチャによって管理されている。インフラストラクチャは、以下の構成要素からなる。

An Object Implementation for Distributed Processing Environment
Youichi TANIBAYASHI, Fumiaki SATO, Tetsuo NAKAKAWAJI, Tadanori MIZUNO
Computer & Information Systems Laboratory, Mitsubishi Electric Corporation

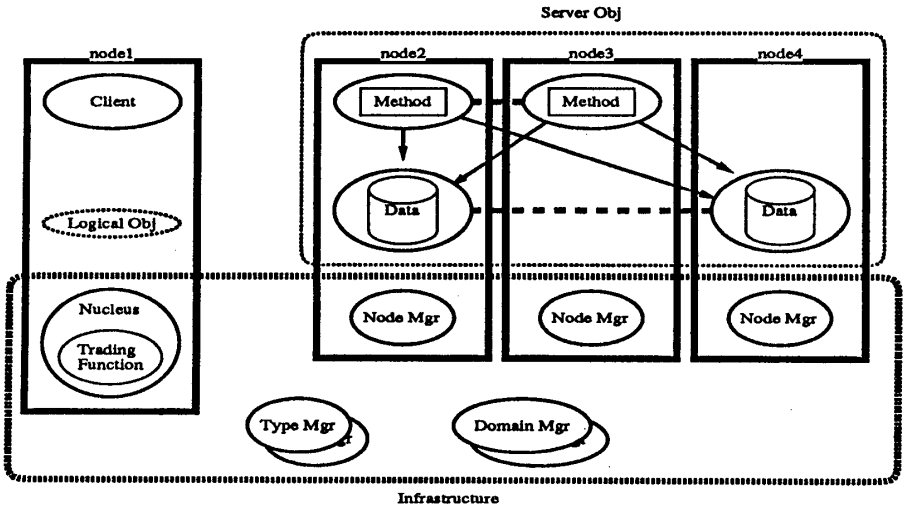

図 1: 手続きとデータが分散したオブジェクトとその管理機構

・ニュークリアス (Nucleus)

各物理ノードに常駐し、クライアントにオブジェクトの分散透過性を提供する。クライアントからの要求に基づき トレーディング ファンクションを用いて最適なサーバ オブジェクトを選択し、そのサーバに対してメッセージを転送する。

・トレーディング機能 (Trading Function)

ニュークリアス内の機能の一つ。クライアントからの要求と タイプ マネージャ、ドメイン マネージャからの情報をもとに 最適なサーバを決定する。

・タイプ マネージャ (Type Manager)

ドメイン内のオブジェクトやクラスに関する名前と オブジェクトとの間の関係、および、クラス名とそのクラス情報を持つノードの関係 を管理している。

タイプ マネージャが管理している名前には、クラスおよびオブジェクトを 一意に識別するプリミティブ (primitive) 名と 性質を表すディスクリプティブ (descriptive) 名がある。

タイプ マネージャは、概念的には、ドメイン内に一つの存在であるが、実際には複製を持つことにより分散されている。

・ドメイン マネージャ (Domain Manager)

ドメイン内の各オブジェクトに関する管理情報を保持している。

各ノード マネージャからの情報に基づいて、オブジェクトが動作しているノード、各ノードの負荷情報などを一元管理している。

ただし、タイプ マネージャと同様に、ドメイン マネージャも複製を持つことができる。

・ノード マネージャ (Node Manager)

各物理ノードに常駐し、オブジェクトの動作状況や 負荷情報をドメイン マネージャに報告する。また、クラス情報も管理しており、オブジェクトの生成削除も行なう。

以上の構成要素が協調動作することにより、複数ノードに分散しているオブジェクトを 利用者には、一つの論理的なオブジェクトとして透過的に見せている。

4 キャッシュによる最適化

本システムの問題点としては、メソッド部とデータ部が分離しているため、データのアクセスの性能が低下する ということが考えられられる。この問題に対する解決策としては、メソッド部にキャッシュを持たせることが考えられる。これにより、常にデータ部へ直接アクセスする必要がなくなり、効率化が行なえる。

逆に、キャッシュを積極的に利用することにより、最適なサービス提供を行なうことが可能である。すなわち、クライアント毎にデータのアクセスに局所性があり、複数のメソッド部がそれぞれ データの別々の部分をキャッシュしている場合、各クライアントが必要としているデータ部分によって メソッド部を振り分け、サービスの最適化が可能である。

5 おわりに

本稿では、処理の分散と サーバの集中的な管理を 統合した 分散環境におけるオブジェクトの実現方式について提案した。

本システムでは、オブジェクトを手続き (メソッド部) とデータ (データ部) に分離し、それぞれを複数のノードに分散配置することにより、単一ノードで保持しきれないような大きなデータ持った オブジェクトを生成可能とし、複雑で大規模な分散環境で最適なサービスを提供できるようにした。さらに、この様なオブジェクトを管理し、システム構成の変化に対しても 透過性を保持したまま柔軟に対応できるようするための インフラストラクチャの構造について述べた。

参考文献

[1] 谷林他、処理分散と資源管理を統合した分散処理環境の提案 (1) — 計算モデル —、第 43 回情報処理学会全国大会 (1991).

[2] 中川路他、処理分散と資源管理を統合した分散処理環境の提案 (2) — 実現機構 —、第 43 回情報処理学会全国大会 (1991).