マルチメディア通信と分散処理　52－17
（1991．9．25）

処理分散と資源管理を協調させた分散処理システム

中川路　哲男　　谷林　陽一　　水野　忠則

三菱電機 (株) 情報電子研究所

あらまし　従来の分散処理システムにおいては、処理の分散と資源の集中管理は、その排反する側面もあるためそれぞれ独立に検討、実現されており、統合されていなかった。これに対して我々は、サーバにおいて，処理のサービスインタフェースと管理操作インタフェースの定義を行い，管理操作により得られる管理情報をサーバの選択に利用することによる、効率や信頼性の高い分散処理システムを提案する。
ここでは、同システムの計算モデルと工学モデル（実現機構）について検討した結果を報告する。

Proposal of a Distributed Processing System in Which Distributed Processing and Resource Management are Integrated

Tetsuo NAKAKAWAJI　Youichi TANIBAYASHI　Tadanori MIZUNO

Computer & Information Systems Laboratory,
Mitsubishi Electric Corporation.

5-1-1, Ofuna, Kamakura-city, 247 JAPAN

Abstract　A distributed processing system in which distributed processing and resource management are integrated is presented. For a server object, service interface and management operation interface should be defined. By utilizing management information of the server objects, the interaction between the client object and the server objects can be optimized. Then the system can provide more reliable and more efficient computing service. This paper presents the computational model and the engineering model of the system.

1 はじめに

マイクロプロセッサ技術の急速な発展による計算機の高性能，低価格化と，ＬＡＮ，ＩＳＤＮなどのネットワーク技術の普及により、システムの処理形態が大型ホストに多数の端末を接続する集中型から、パーソナルコンピュータまたはワークステーションをネットワークで接続する分散型に変化しつつある。分散型のシステムでは、資源を共有したり，処理を各計算機に分散させることができるため、コストパフォーマンス，拡張性や信頼性にの高いシステムを構築することが可能となる．

一方、分散システムの大規模化、複雑化に伴い、システム内には、多くの資源が分散して存在するため，それらを集中的に管理／運用していくことが重要である．

従来の分散処理システムにおいては、これらの処理の分散と資源の集中管理は、その排反する側面もあるためそれぞれ独立に検討、実現されており、統合されていなかった。これに対して我々は、管理情報をサーバの選択に利用することによる、効率や信頼性の高い分散処理システムを提案する。

ここでは、そのソフトウェアの側面から見た計算モデルと，それを実現するための機構の側面から見た工学モデルについて検討した結果を報告する。

2 計算モデル

2.1 提案モデル

分散システムは、資源と処理が分散された処理形態である。その意味で，資源を内包してその資源へのアクセス機能をサービスするオブジェクトの集合としてモデル化することができる。オブジェクト間の相互作用の結果，システムとしての情報処理が遂行される。一つのオブジェクト間の相互作用は、他のオブジェクトに特定のサービスを提供するサーバオブジェクトと，そのサービスを利用するクライアントオブジェクト間のメッセージ交換と考えられる。オブジェクトは，提供するサービスの仕様，すなわちメッセージの形式を記述したクラスの実行主体である。

一方，オブジェクトを分散させただけでは，効率や信頼性の高い分散システムを運用することはできない。オブジェクトの名前，位置や状態を管理し，セキュリティを確保するための監視を行い，必要に応じてオブジェクトの生成／消去や試験を行う必要がある。分散システムにおけるオブジェクトの管理のために，様々なアーキテクチャとプロトコルが提案されている [1]。

今回我々が提案するのは，これらの処理分散と資源管理を統合した分散処理システムである。ソフトウェア（プログラム）から見た本システムのモデル（計算モデル）を図図 1に示す。

図 1の計算モデルの構成要素を以下に示す。

- サーバオブジェクト

  サーバオブジェクトは、クライアントオブジェクトに対してサービスを提供するオブジェクトである。また、インフラストラクチャに対して、サーバオブジェクトを管理するための操作も提供する。サービスおよび管理操作は、それぞれサービスインタフェースおよび管理操作インタフェースを通して行なわれる。

- クライアントオブジェクト

  クライアントオブジェクトは、サーバオブジェクトからあるサービスを受ける際に、まず、インフラストラクチャのトレーディングインタフェースを通して最適なサーバオブジェクト（群）を問い合わせる。インフラストラクチャから最適なサーバオブジェクト（群）を得たクライアントオブジェクトは、そのサーバオブジェクトのサービスインタフェースを通して実際のサービスを得る。

- インフラストラクチャ

  インフラストラクチャは，これらのオブジェクトを

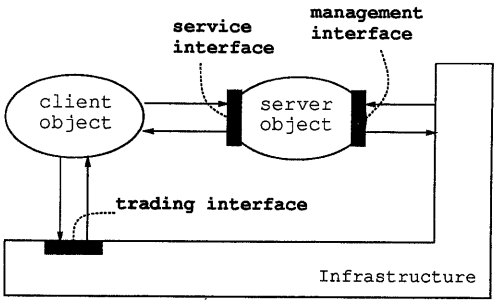

図 1: 計算モデル

管理する主体である。計算モデルでは，インフラストラクチャの提供する位置透過性により、各オブジェクトは物理的な位置は意識する必要がない。クライアントオブジェクトがサービスを要求するサーバオブジェクトを指定する際には、インフラストラクチャのトレーディングインタフェースを利用する。インフラストラクチャは、管理情報を基に最適なサーバオブジェクトを選択する。サーバオブジェクトの管理情報は、サーバオブジェクトの管理操作インタフェースを通して取得する。

2.2 名前

本システムでは，以下の 4 種類の名前を定義している。

- クラスの primitive 名：クラスを一意に識別する名前。
- オブジェクトの primitive 名：オブジェクトを一意に識別する名前。
- クラスの descriptive 名：クラスの性質を記述する名前。一つの名前に対して複数のクラスが適合し得る。また，それらのクラスから生成されたオブジェクトは，全てこの名前に適合する.
- オブジェクトの primitive 名：オブジェクトの性質を記述する名前。一つの名前に対して複数のオブジェクトが適合し得る。

2.3 オブジェクトのインタフェース定義

分散システム，特に異機種計算機を結合した分散システムでは，それらの間のインタフェースを明確に定義しておくことが重要である。本システムには、以下の 3 種類のインタフェースが存在する。持つ。

1. サービスインタフェース

   サーバオブジェクトが提供するサービスのインタフェースである。クライアントオブジェクトは、サーバオブジェクトから export されたこのインタフェース仕様に従って、サービスを要求する。これには、以下のような情報が含まれる。

   - サービスの名前、または識別子
   - サービスを要求するために必要な付加情報（引数とその型）
   - サービスを実行した結果の型ととり得る値

   これらはいわゆる RPC(Remote Procedure Call)[2] におけるインタフェース定義に相当するものである。

2. 管理操作インタフェース

   サーバオブジェクトを管理するための操作のインタフェースである。このインタフェースは、インフラストラクチャから使用される。このインタフェースには以下のような情報が含まれる。

- 生成／消去のためのインタフェース
  生成時の付加情報（引数とその型）や生成時に付与されるオブジェクトの primitive 名などの情報を含む。
- 属性獲得／設定のためのインタフェース
  サーバオブジェクトの位置、状態、負荷、障害履歴、アクセス制御リスト、サービス実行のための時間とコストなどの各種属性を獲得したり、設定するためのインタフェース。サービス実行のための時間とコストには、実際に処理に必要となるデータの転送時間とコストと時間が含まれる。
- 事象通知のためのインタフェース
  障害、不法アクセス、各種属性変更などの通知をサーバオブジェクトからインフラストラクチャに行なうためのインタフェース。
- 試験／計測のためのインタフェース
  サーバオブジェクト内で、自己診断試験や負荷計測、応答時間計測を行なわせるためのインタフェース。

3. トレーディングインタフェース
   クライアントオブジェクトが、受けようとするサービスを最適の条件で提供してくれるサーバオブジェクトを知るための情報を指定するインタフェースである。トレーディングインタフェースには、以下のような情報が含まれる。
   - サーバオブジェクトの名前
   - 受けようとするサービス名
   - 選択条件 (例えば、負荷の最も軽いサーバオブジェクト、コストの最も安いサーバオブジェクト、利用可能な全サーバオブジェクト、サービスを規定時間内に完了するサーバオブジェクトなど)

3 工学モデル

3.1 提案モデルとインフラストラクチャの構造

1章で提案した計算モデルを実現する工学モデルを図2に示す。

図2において、クライアントオブジェクトに対して処理分散と資源管理を統合したサービスを提供するインフラストラクチャの各構成要素の概要を以下に述べる。

- ニュークリアス
  各物理ノード毎に存在し、オブジェクトに分散透過性を提供する機能である。そのために、トレーディングインタフェースを通じて要求されたサービス内容の品質に基づいて、トレーディング機能を利用してサーバオブジェクトを選択したり、サーバオブジェクトの存在するノードのニュークリアスと通信して、オブジェクト間の通信の支援を行なう。
- タイプサーバ
  タイプサーバは、クラスの primitive 名と、そのクラスが提供するサービス名、そのクラスから生成されたオブジェクトの primitive 名などのクラスに関する情報を保持し、それらの情報に関する検索サービスを提供する機能である。例えば、クラスの primitive 名からそのクラスのオブジェクトの primitive 名を提供したり、クラスの descriptive 名を解析して適合するクラスの primitive 名を提供したり、サービス名からそのサービスを提供するクラス名を提供する。descriptive 名の解析に動的な情報（管理情報）が必要な場合には、ドメインマネジャにアクセスを行なう。
- ドメインマネジャ
  ドメインのポリシーに基づいて、ドメイン内のサーバオブジェクトの管理を行なう機能である。各サーバオブジェクトの位置、状態や負荷などの管理情報を仮想的なデータベースに蓄積して、オブジェクト

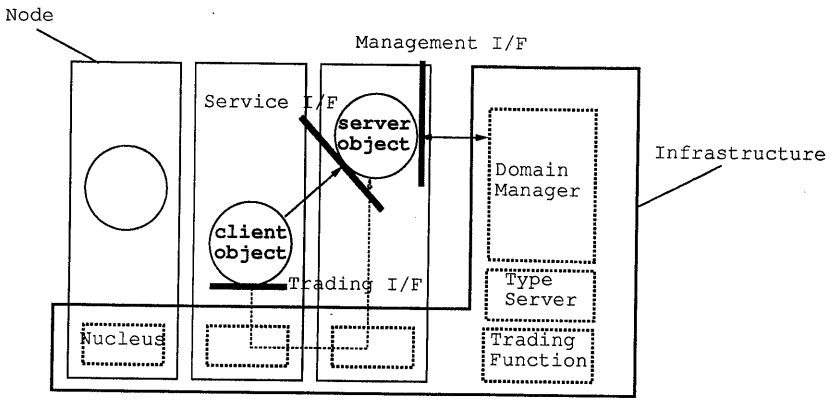

図 2: 工学モデル

の primitive 名で管理する。管理情報の蓄積は、サーバオブジェクトに対して管理操作インタフェースを通したアクセスを行なうことにより、データベース内の管理情報を更新する。ドメインマネジャ内の管理情報に基づいて、トレーディング機能はサーバオブジェクトの選択を行なう。

また、ドメインマネジャは、ドメイン内のサーバオブジェクトの状況を監視し、必要に応じてサーバオブジェクトの生成／消滅も行なう。生成時には、サーバオブジェクトで定義された管理操作インタフェースを用いて、適切なノードにサーバオブジェクトを生成し、名前を付与する。

- トレーディング機能 [3]

  トレーディング機能は、クライアントオブジェクトから指定されたトレーディングインタフェースを基に、最適なサーバオブジェクト（群）を選択する。選択に当たっては、descriptive 名を解析したり、サービス名からクラスやオブジェクト名を引き出すためにタイプサーバを、サーバオブジェクトの管理情報を得るためにドメインマネジャを利用する。

3.2 実現例

サーバオブジェクト選択の例を以下に示す。

1. クライアントオブジェクトは、トレーディングインタフェースに基づいてサーバオブジェクトとのバインド要求を発行する。この要求を実際に受けるのは、クライアントの存在するノードのニュークリアスである。

2. ニュークリアスは、クライアントオブジェクトから受けた要求をトレーディング機能に転送する。

3. トレーディング機能は、まず名前解析を行なう。指定されたサービスを提供しているサーバオブジェクトのクラスを、タイプサーバに依頼して検索する。同時にクラス名の descriptive 名が指定されている場合には、その解析もタイプサーバに依頼する。タイプサーバからは、指定したサービスを提供しているオブジェクト名のリストが返される。

4. オブジェクト名のリストを得たトレーディング機能は、それらのオブジェクトの管理情報をドメインマネジャに要求する。ドメインマネジャからは、各サーバオブジェクト毎に、位置や情報、負荷、課金情報、アクセス権情報、障害情報、版名などの管理情報が返される。

5. トレーディング機能は、まずクライアントオブジェクトにそのサービス要求を行なう権利があるかとい

うことと、サーバオブジェクトが活動状態にあり、新たなサービスを提供する余裕があるかをチェックし、サーバオブジェクトの候補を絞る。次に、サービスを実行するためのコストと時間を見積もり、クライアントの要求したコスト内でサービスが提供されるかということと、クライアントが要求した時間内でサービスが終了するかという観点から各サーバオブジェクトを評価する。この時間の計算には、クライアントオブジェクトとサーバオブジェクト間の通信時間、サーバオブジェクトでの待ち時間、サーバオブジェクトでのサービス時間が考慮される。評価の結果最も評価点数の高いサーバオブジェクトが最適サーバオブジェクトとなり、ニュークリアスにその名前と位置が返却される。

6. サーバオブジェクトの名前と位置を受け取ったクライアントノードのニュークリアスは、そのサーバオブジェクトの存在するノードのニュークリアスと通信を行ない、クライアントとサーバオブジェクトを結合する。

3.3 サーバオブジェクト選択アルゴリズム

理想的には、上記のようなトレーディング機能における評価で、最適なサーバオブジェクトが選択される。しかし実際には、ドメインマネジャとサーバオブジェクト間の通信が通信路の負荷を増長したり、古い管理情報を利用することによる特定サーバオブジェクトへの集中アクセスが憂慮される。

これを解決するために本システムでは、以下のようなサーバオブジェクト選択アルゴリズムを使用する。

- 管理目的の通信用に一定のチャネルを確保する。また、管理情報収集のための通信は通信路の負荷が一定以上の時は行なわない。負荷や状態などの動的な管理情報に関しては、ドメインマネジャからサーバオブジェクトにアクセスするのではなく、変化があればサーバオブジェクトから通知する方向で、通信路の負荷を減らす。
- ドメインマネジャは、各ニュークリアス内の通信処理部分も管理対象として管理することにより、各ノードの通信負荷状況を把握する。
- 負荷の監視には、上昇方向の閾値と下降方向の閾値の2種類を利用する。
- 最新の管理情報が不足している時は、サーバオブジェクトの選択に乱数的な要因を入れる。

4 おわりに

本稿では、処理の分散と資源の集中的な管理を統合した分散処理システムを提案した。今後は、プロトタイプの実装を通じてより詳細な検討を行なう予定である。

参考文献

[1] ISO/IEC 10040, *OSI System Management - Overview*、ISO, 1991.

[2] A.D.Birell and B.J.Nelson、*Implementing Remote Procedure Calls*、ACM Trans. on Computer Systems、Vol.2、pp.39-59.

[3] ISO/IEC JTC1/SC21/WG7 N312, *Working Document on the Trader*, ISO, 1990.