

ODP参照モデルの分散型通信システム構築への適用に関する考察

3M-2

藤長 昌彦 加藤 聰彦 鈴木 健二

KDD 研究所

1. はじめに

分散処理技術の進展と普及に伴い、ISOとCCITTは、分散処理の標準化を目的としてODP (Open Distributed Processing: 開放型分散処理) の検討を行なっている。現在、汎用的な分散処理技術の機能を抽出、整理し、分散処理システムのモデル化を行なうための枠組みを規定したODP基本参照モデルの標準化を進めている^[1,2]。一方、メッセージ通信処理やインテリジェント・ネットワーク等の高度な通信システムの構築においては、複数の計算機システムを使用して機能分散や負荷分散を行なう分散型の構成が提案されている^[3,4]。このような分散型通信システムでは、多数の構成要素が協調動作を行なうため、その構築に当たっては体系的な技法を採用することが望ましい^[5,6]。従って、分散処理の枠組みを与えるODP基本参照モデルを適用し、分散型通信システムをトップダウン的に設計するアプローチが有用と考えられる。本稿では、ODP基本参照モデルを用いた分散型通信システムの構築について考察した結果を述べる。

2. ODP基本参照モデルの概要

ODP基本参照モデルでは、分散処理を用いた任意の情報処理システムをモデル化するための概念を定めた記述モデル (Descriptive Model) と、分散処理システムを「開放型」と呼べるための性質を定めた規定モデル (Prescriptive Model) を規定している。分散処理システムを抽象化するための枠組みとして以下の五つの視点 (ビューポイント) を導入し、各ビューポイントから見たシステムのモデルを記述するための概念や規則、構造等を規定するビューポイント言語を定めている。

エンタプライズ・ビューポイント：情報処理システムの全体的な目的をモデル化する視点。管理の方針やシステムの動作環境等が記述される。

情報ビューポイント：情報処理システムの扱う情報をモデル化する視点。情報の構造、情報要素間の関係、情報の流れや処理に伴う変化が記述される。

コンピューショナル・ビューポイント：情報処理システムの処理機能とデータ型に着目してシステムを機能的に分割し、その構成要素をモデル化する視点。構成要素間の通信機能や、構成要素の分散を隠蔽する分散透過性等を提供するインフラ

ストラクチャの概念が導入される。構成要素はインフラストラクチャの機能を利用して設計され、構成要素間の関係や並列動作が規定される。

エンジニアリング・ビューポイント：計算機のオペレーティングシステム (OS) やプロセスの内部構造をモデル化する視点。計算機、OS、アドレス空間等を抽象化した概念が導入される。システムの性能を考慮し、分散透過性の実現機構やプログラムやデータの分散の仕方等を記述する。

テクノロジー・ビューポイント：情報処理システムを実装する方法を記述する視点。

3. ODP基本参照モデルを用いた分散型通信システム構築のアプローチ

分散型通信システムの構築においては、開発の容易性や、機能や処理能力の拡張性を高めるために、分散処理技術等の汎用技術を用いた実装基盤 (プラットフォーム) を導入し、その上に通信処理を実現するアプリケーションを作成するというアプローチが有用である。

プラットフォームの機能を明確化するためには、分散型通信システムのコンピューショナル言語を用いたモデルを規定する必要がある (図1参照)。プラットフォームは、分散を意識させないためのサポート環境であるインフラストラクチャと、通信システムの機能の内、共通的に括りだせる通信システム用共通機能から構成される。このモデルに基づいてプラットフォームの機能やインタフェースを決定し、通信処理アプリケーションの開発方法を検討する。

一方、プラットフォームの実現手法を検討するためには、エンジニアリング言語を用いたモデルを規定する必要がある (図2参照)。このモデルを用いて、分散透過性を実現するトランスペアレンシ機能、通信システム用共通機能の実現方法、複数の計算機上における通信処理アプリケーションの配置等について、システム性能を考慮しつつ検討する。コンピューショナル・モデルで規定された機能を実現できない場合や、要求される性能が達成できない場合にはコンピューショナル・モデルの再設計を行なう

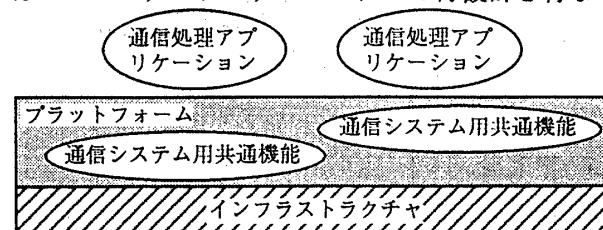


図1 分散型通信システムのコンピューショナル・モデル

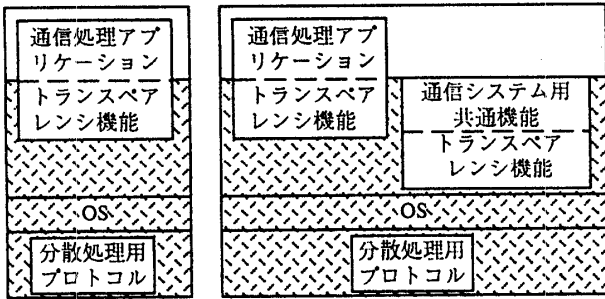


図2 分散型通信システムのエンジニアリング・モデル

必要もある。

4. 分散型MHSシステムへの適用例

ここでは、複数の計算機をLANで結合した分散型のMHS (Message Handling System) システムの設計に、前章で述べたアプローチを適用した例について述べる。分散型MHSシステムの設計の第一段階において、どのようなプラットフォームを導入するかを決定する必要がある。ここでは以下の二つの例を考える。

- ① クライアントサーバ・モデルに基づくRPCを用いた既存の分散処理環境をプラットフォームとして採用する場合
- ② MHSシステムのメッセージを自律的なオブジェクトとして扱うオブジェクト指向分散環境をプラットフォームとして新たに導入する場合

4.1 既存の分散処理環境を用いた設計例

図3に示すように、通信処理アプリケーションはMHSのメッセージの処理を行なうMHSサーバや、ルーティング情報を管理するルーティングデータベースサーバ等の、RPCにより相互に通信しあうサーバとして実現される。

プラットフォームでは、これらのサーバに対してRPCのインタフェース記述言語によるプログラミングインタフェースを提供する。また、プラットフォームは、受信したMHSのメッセージや、メッセージ処理に関するサーバの内部状態等の、紛失が許されないデータを保持するログサーバを準備するとともに、関連する複数のログを一括してログサーバに蓄積する機能を提供する。更に、システムの可用性と処理能力を高めるために、MHSサーバのレプリケーションを行ない、MHSサーバの負荷を監視し、最も負荷の軽いMHSサーバを選択する機能を提供する。

次に、エンジニアリング・モデルを用いてプラ

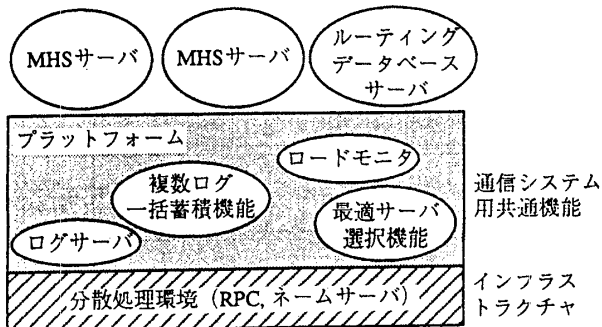


図3 既存の分散処理環境を用いたプラットフォーム

トフォームの実現方式を検討する必要があるが、この場合は、既存の分散処理環境を前提としているため、RPCを利用した通信システム用共通機能の実現法に着目するのみでよい。

4.2 オブジェクト指向分散処理環境を導入した設計例

オブジェクト指向分散処理環境を新規に実現する際には、図4に示すように、MHSシステムで扱う個々のメッセージや、ルーティング情報を保持するデータベース等が自律的なオブジェクトとなる。この場合、インフラストラクチャとしてのオブジェクト指向分散処理環境そのものがプラットフォームとなり、クラス定義、インスタンスの生成、オブジェクト間通信及び実行制御等の機能を提供する。更に、システム障害の発生に対処するために、オブジェクトの状態を必要に応じてチェックポイントし、オブジェクトが異常終了した場合、自動的に新しいインスタンスを生成して処理を継続させる「パーシステントなオブジェクト」をサポートする。

このようなプラットフォームを実現するためには、エンジニアリング・モデルにおいて、粒度の細かいオブジェクトの実行環境を設計する必要がある。例えば、既存のRPCを用いる分散処理環境上に開発する場合には、オブジェクトの実行を、スレッド機能を持つオブジェクト処理サーバにより実現し、オブジェクト間通信やパーシステントな情報の蓄積を、RPCやログサーバにより実現する等の手法を検討する必要がある。

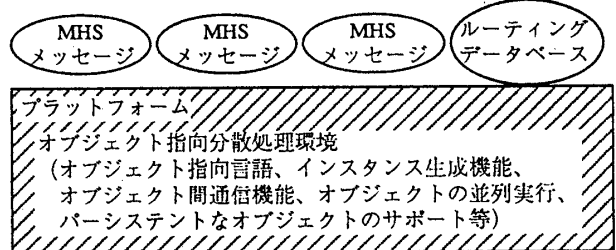


図4 オブジェクト指向分散処理環境を用いたプラットフォーム

5. むすび

本稿では、ODP基本参照モデルに基づき、コンピュータとエンジニアリングの二つのビューポイントに着目して、分散型の通信システムを体系的に構築するアプローチを示した。最後に、日頃御指導戴くKDD研究所小野所長、浦野次長に感謝する。

参考文献

- [1] ISO, "Interim Revised Text of CD 10746-2, Basic Reference Model of ODP - Part 2: Descriptive Model," ISO / IEC JTC1 / SC21 N6767, February 1992.
- [2] ISO, "Draft Recommendation X.903: Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive Model," ISO / IEC JTC1 / SC21 N7055, June 1992.
- [3] 加藤, 藤長, 鈴木, "分散処理技術を用いた通信システムの構築に関する一考察," 情処学会第43回全大, 7T-1, 1991年10月.
- [4] Fleck, et al., "The INA Architecture: An Architecture for Information Networks," Proc. of TINA 92, 13-1, January 1992.
- [5] Fujinaga, Kato and Suzuki, "Implementing IN Functional Entities on top of Distributed Operating System," Proc. of ISS'92, Vol. 1, P22, pp. 268 - 272, October 1992.
- [6] 加藤, 小林, "情報通信アーキテクチャの提案," 第3回情報通信ネットワークアーキテクチャワークショップ 資料, 1991年10月.