

分散アプリケーションの実行環境に関する検討

2W-7

李相詰 松方純† 浅野正一郎††

(†東京大学 ††宇宙科学研究所 ††学術情報センター)

1 はじめに

分散処理システムにおいては、複数のモジュールが相互に協調動作しながら処理を行なう分散アプリケーションが、システム資源の有効利用などの面で望ましいと考えられる。しかし、分散アプリケーションの構築は、開発者にとって負担が大きことが多い。

本稿では、モジュールの位置透過性及び、協調動作に有効な一対多通信を考慮することによって、分散アプリケーションの構築・実行の容易化を図るためのシステムを提案する。提案するシステムでは、モジュールに付加される属性やネットワーク一意な名前を用いて、位置透過な通信を実現する。さらに、一対多通信においても位置透過性を考慮することによって、種々の分散アプリケーションに対する応用が考えられる。

2 提案するシステムの概要

本節では、分散アプリケーションを構成する基本単位であるモジュールについて触れ、その後、提案するシステムとその動作について述べる。

2.1 モジュール

分散アプリケーションは、基本単位であるモジュールから構成されており、ネットワーク一意な名前と、属性が付加されているものとする。また、起動前のモジュールをテンプレート、起動後の実行可能な状態にあるモジュールをインスタンスということにする。さらに、同一の属性を持つモジュールの集合をモジュールグループと定義する。

2.2 システムの構成

提案するシステムは、Communication Manager, Transparency Manager, Module Manager から構成される。システム概念図を図1に示す。

各Managerの主な機能は以下の通りである。

1. Communication Manager

- 通信における受信モジュールの決定
- 通信メッセージの送信
- 一対多通信において、メッセージの分配及び受信モジュールが通信メッセージを受信したことの確認など

2. Transparency Manager

- モジュールに関する情報(モジュールの位置、モジュール名、モジュール属性など)を管理

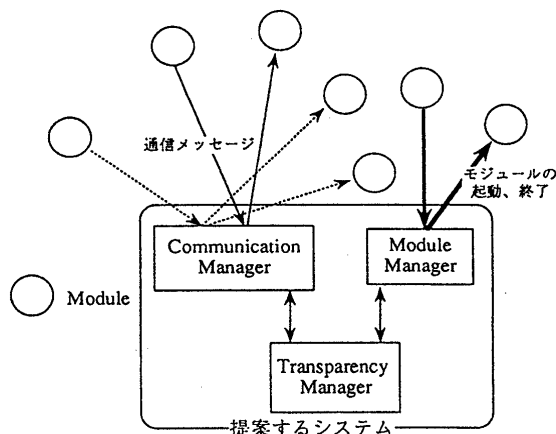


図1: 提案するシステム概念図

- ハードウェア資源とその状態に関する情報(計算機の名前・種類・使用状況など)を管理

3. Module Manager

- モジュールの起動や終了に関する操作

2.3 システムの動作

提案するシステムの、主な動作は以下の通りである。

まず、モジュール起動時には、起動要求を受けたModule Managerが、起動すべきモジュールの情報をTransparency Managerに問い合わせ、その問い合わせ結果(テンプレートの位置、そのモジュールを起動可能な計算機など)に基づいてモジュールを起動する。さらに、起動したモジュールの情報(位置など)をModule ManagerがTransparency Managerに通知することによって、モジュール情報を動的に管理することができる。

モジュールの終了時は、起動時と同様に、終了要求を受けたModule Managerが、そのモジュールに関する情報をTransparency Managerに問い合わせ、その結果を用いて終了動作を行なう。そして、そのモジュールが終了したことをTransparency Managerに通知する。

また、モジュール間の通信については、送信モジュールが、名前や属性によって受信モジュールを指定し、Communication Managerに送信要求を出す。Communication Managerは、受信モジュールの情報をTransparency Managerに問い合わせ、その結果を用いてメッセージを送信する。一対多通信の場合には、指定され

た属性を持つモジュール全てに、メッセージを分配して送信する。

アプリケーションの起動や終了については、そのアプリケーションを構成するモジュールの情報を所有しアプリケーションの初期化や終了などを行なうメインモジュールを考える。そして、このメインモジュールが、アプリケーションを構成するモジュールの起動・終了要求を Module Manager に対して出すことによって、アプリケーションの起動・終了などの動作を行う。

2.4 特徴

提案するシステムの特徴としては、モジュールの位置透過性の実現とその動的な管理、モジュール間通信における属性を用いた受信モジュールの指定と一対多通信機能による、開発の負担軽減などが挙げられる。

なお、関連する研究として、Frame Works system[1]が挙げられる。このシステムは、プロセス（モジュール）の位置透過性の実現と、プロセス間通信を容易に利用可能とすることによって、分散アプリケーションの構築を支援している。しかし、一対多通信が困難であることや、プロセス情報の管理が静的であるという制約がある。

3 Manager について

Module Manager と Communication Manager は各計算機毎に存在する。また、Transparency Manager は、ネットワークに 1 個存在するものとする。ここでは、Transparency Manager が管理する情報と、モジュール起動時の Module Manager の動作について述べる。

3.1 Transparency Manager

Transparency Manager は、次の情報を管理する。

モジュール（テンプレート）の情報 モジュール（テンプレート）の名前、位置、そのモジュールを起動可能な計算機の種類、機能名などの属性を管理する。

計算機資源とその状態 計算機の名前、種類、速度、状態を管理する。速度については、基準となる計算機との相対速度を用いて表現するという方法が考えられる。状態は、その計算機が利用可能かどうかを表す。

モジュール（インスタンス）の情報 モジュール（インスタンス）の名前、位置、モジュールのテンプレートの名前を管理する。

3.2 Module Manager

Module Manager は、各計算機上におけるモジュールの起動や終了を管理する。

モジュール起動時に、Module Manager は、起動するモジュールのテンプレート名を用いて、Transparency Manager に情報を問い合わせる。Transparency Manager は、そのモジュールを起動可能な種類であり、かつ、利用可能な計算機（一般に複数）の名前や速度などの情報を、モジュールテンプレートの位置と共に返す。

Module Manager は、モジュールを起動する計算機を Transparency Manager によって与えられた中から

選択する。計算機の実験方法については、最も高速なものを選ぶ、という方法などが考えられる。

次に、Module Manager がモジュールを起動し、そのインスタンスに名前を付ける。但し、遠隔の計算機をモジュール起動先として選択した場合には、その計算機上の Module Manager にモジュールの起動及びインスタンスの名前付けを依頼する。

最後に、Module Manager は、起動したモジュールのインスタンスの位置や名前、テンプレート名を Transparency Manager に通知する。Transparency Manager は、その情報を格納する。

なお、Transparency Manager にはアクセスが集中する傾向にあるため、大規模なシステムに適用する際には、機能毎の分散化やキャッシングなどを行なう必要があるであろう。

4 応用例

提案するシステムの応用として、次のような例が考えられる。

等しい機能を提供する独立した複数のモジュールが同一のモジュールグループに属している場合、そのモジュールグループに対して通信を行うことによって処理を行なわせる方式が考えられる。この場合、呼び出し側は、特定のモジュールの実体を意識する必要はなく、モジュール属性によって表される機能そのものを見ることになる。

また、耐障害性向上への応用も考えられる。例えば、Module Manager が、同一のアプリケーションを異なる計算機上で複数個実行させ、最初に終了するアプリケーションの結果を採用する、という方式にも適用できる。

5 おわりに

本稿では、モジュールの位置透過性の実現と一対多通信を考慮することによって分散アプリケーションの構築・実行の容易化を図るシステムを提案した。また、動的なモジュールの管理を行うことも提案するシステムの特徴の一つである。

今後の課題としては、属性の具体的な内容、モジュールの動的な管理方法等の検討が挙げられる。

また、提案するシステムは開発者の視点に基づいているが、さらに、サービス利用者の視点も考慮した環境を考えることが出来る。例えば、サービスという機能的な要素以外にも、実行性能や耐障害性などというサービスの品質（非機能的な要素）もサービス利用者が指定し、そのサービス品質を満たすように、システムがアプリケーションを動的に管理する、という方法も考えられる。このためのモデルの考案や評価に関しても、今後の課題に含めることができよう。

[参考文献]

- [1] A. Singh, J. Schaeffer and M. Green, "A Template-Based Approach to the Generation of Distributed Applications Using a Network of Workstations," *IEEE Trans. Parallel and Distributed Systems.*, vol. 2, No. 1, pp. 52-67, Jan. 1991.