

アドホック分散システムのための資源管理手法

藤田 耕作[†] 瀧本 栄二[†] 毛利 公一^{††} 大久保 英嗣^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学理工学部情報学科

1 はじめに

複数のパーソナルコンピュータや携帯情報端末で構成され、それらが動的に移動、接続、切断されるシステムでは、システム構成が動的に変化する。このようなシステムに適用可能なオペレーティングシステム（以下、OS）として、分散 OS AG[1]を開発している。AGは、OSの構成要素であるエージェントが自律的に動作し、複数のエージェントと協調することにより、環境の変化に動的に適応することが可能である。AGでは、計算機資源をオブジェクトとして抽象化し、エージェントがオブジェクトを操作することにより、資源管理を行う。

従来のOSでは、ファイル管理を始めとした資源管理を木構造に基づいて行う。特に、ファイルは、2次記憶装置に木構造に基づいて格納されるため、ファイル間に物理的依存関係が存在する。多くの分散システムでは、システム全体で1つの木構造を構成することにより、位置透過なファイル管理を行う。このため、1つのリンクが消えることにより、多くのファイルへのパスが失われる可能性がある。本稿で対象とするような動的に構成が変化する分散システムでは、このようなファイル間の物理的依存関係は望ましくない。

本手法では、データや周辺デバイスといった2次記憶装置に格納されるオブジェクトを対象として、それぞれのオブジェクト間で物理的依存関係を持たないフラットな構造を用いた管理を行う。本手法により、システム構成の変化に柔軟に対応することが可能となる。

以下、本稿では、2章で適用の対象となるAGの概要、3章で従来の資源管理手法について述べる。4章で本手法の特徴、5章でまとめと今後の課題について述べる。

2 AGの概要

AGは、マイクロカーネル方式を基本としており、計算機資源を抽象化するオブジェクト群、カーネルとしての機能を実現するエージェント群から構成される。オブジェクトは、内部状態および内部状態を変化させるためのメソッドを持ち、エージェントに対して統一的な操

作インタフェースを提供する。データ、周辺デバイスを抽象化するオブジェクトは、2次記憶装置に格納される。エージェントは、カーネルを機能単位で分割し、それぞれを1つのプロセスとして実現したものである。AGでは、複数のエージェントが協調することにより、OSとしての機能が実現される。資源管理は、エージェントがオブジェクトのメソッドを呼び出すことにより行う。エージェントは、協調する他のエージェントや自身が管理するオブジェクトといった環境を監視し、その環境に最適な処理を行う。これにより、環境の変化に動的に適応することが可能である。

3 従来の資源管理手法

従来のOSでは、データや周辺デバイスといった資源をファイルとして抽象化し、ユーザに対して統一的な操作インタフェースを提供している。最も一般的なファイル管理手法である階層的ファイルシステムでは、ファイルをディレクトリに格納し、ディレクトリを木構造で管理する。分散システムにおいて、位置透過な資源管理を行うためには、それぞれの計算機が保持するディレクトリツリーが統一されていなければならない。しかし、分散システムでは、ファイルやディレクトリを格納する2次記憶装置は分散配置される。このため、従来の分散ファイルシステムでは、2次記憶装置のパーティションをディレクトリツリー上の任意のディレクトリにマウントすることにより、複数の計算機に接続されている2次記憶装置を、1つのディレクトリツリーで管理することを可能としている。ディレクトリツリーには、以下の2つの依存関係が存在する。

- ディレクトリとディレクトリ間、ディレクトリとファイル間の論理的依存関係
- ディレクトリと2次記憶装置間の物理的依存関係

これらの依存関係が存在するため、2次記憶装置の動的な切放しによるディレクトリツリーの変更に柔軟に対応することが困難である。

4 AGにおける資源管理手法

本稿では、データや周辺デバイスといった2次記憶装置に格納されるオブジェクトを、それぞれのオブジェクト間で物理的依存関係を持たないフラットな構造により管理する手法を提案する。オブジェクトは、2次記憶装置

A Resource Management Scheme for Ad hoc Distributed Systems

Kosaku Fujita[†], Eiji Takimoto[†], Koichi Mouri^{††}, and Eiji Okubo^{††}

[†]Graduate School of Science and Engineering, Ritsumeikan University

^{††}Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

にフラットに格納される。エージェントは、位置情報を含まないオブジェクト名を指定して、オブジェクトを利用する。以下、本章では、本手法の特徴について述べる。

4.1 オブジェクトの識別

本手法では、対象とする分散システム全体で一意的な値を持つオブジェクト ID を用いて、オブジェクトを一意的に識別する。オブジェクト ID は、オブジェクトの生成時に付与される。オブジェクト ID は、以下に示す情報から構成される。

- 計算機 ID

対象とするシステム上の計算機を一意的に識別する値を持ち、オブジェクトが格納されている計算機を表す。

- ディスク ID

単一計算機上の 2 次記憶装置を一意的に識別する値を持ち、オブジェクトが格納されている 2 次記憶装置を表す。計算機 ID とディスク ID を組み合わせることにより、システム上の 2 次記憶装置を一意的に識別する。

- ローカルオブジェクト ID

2 次記憶装置に格納されているオブジェクトを、その 2 次記憶装置内で一意的に識別する値を持ち、オブジェクトが格納されている 2 次記憶装置内での位置を表す。

エージェントが指定したオブジェクト名は、対応するオブジェクト ID に変換され、オブジェクトの物理的な位置が特定される。

4.2 オブジェクトの検索

エージェントは、オブジェクトを利用する前に、利用したいオブジェクトのオブジェクト ID を取得する必要がある。エージェントは、オブジェクト名をキーワードとして、オブジェクト名とそれに対応するオブジェクト ID の組 (以下、名前情報) を保持するオブジェクトに問い合わせる。このとき、指定した名前のオブジェクトが複数存在する場合は、エージェントまたはオブジェクトを管理するシステムがオブジェクトの種類、作成者といったオブジェクトの属性により最適なオブジェクトを選択する。また、エージェントがオブジェクト名を知らない場合は、自身が属しているサブシステム上のオブジェクトの一覧表示から、オブジェクトの属性などにより利用したいオブジェクトを選択する。

4.3 名前情報の管理

AG では、対象とする分散システムを複数のサブシステムに分割する (図 1)。サブシステムは、任意の計算機

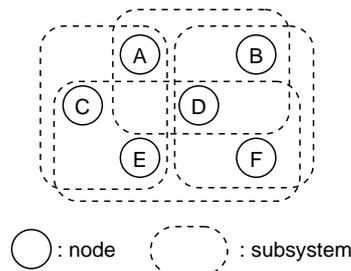


図 1 分散システムの構成

から他の計算機を経由してすべての計算機に到達するように構成される。

それぞれの計算機は、自計算機上の 2 次記憶装置に格納されているオブジェクトの名前情報を保持する。さらに、自身が属しているサブシステム内の他の計算機上の 2 次記憶装置に格納されているオブジェクトの名前情報の複製を保持する。図 1 の計算機 A は、自計算機上のオブジェクトの名前情報および計算機 B, D, C, E 上の名前情報の複製を保持する。このため、計算機 A は、計算機 A, B, D, C, E 上の 2 次記憶装置に格納されているオブジェクトの名前情報を保持することになる。計算機 A において計算機 F 上のオブジェクトの取得要求を受けたときは、同一サブシステム内の計算機 B, D, C, E に取得要求を転送し、目的のオブジェクトの名前情報を取得する。この処理を繰り返すことにより、システム上のすべてのオブジェクトの名前情報を取得することが可能となる。このとき、キャッシュを用いることにより、計算機間の通信を減少することが可能となる。サブシステム単位で名前情報を管理することにより、オブジェクトの取得要求をシステム上のすべての計算機に送信する必要がなく、効率的な名前情報の管理が可能となる。

5 おわりに

本稿では、AG における資源管理手法について述べた。本手法では、2 次記憶装置に格納されるオブジェクトに対して、システム全体で一意的な値を持つオブジェクト ID を付与し、それぞれのオブジェクト間で物理的依存関係を持たないフラットな構造により管理する。これにより、システム構成の変化に柔軟に対応することが可能となる。今後の課題としては、オブジェクトを他の 2 次記憶装置に移動したときの管理方法、計算機間の名前情報の同期方法についての検討が挙げられる。

参考文献

- [1] 瀧本栄二, 芝公仁, 大久保英嗣: エージェント技術を用いた分散オペレーティングシステムの構成手法, 情報処理学会研究報告 2002-OS-89, Vol. 2002, No. 13, pp. 117-123 (2002).