

広域分散環境における資源管理

村井純
東京大学

田中啓介
上智大学

計算機ネットワーク技術により接続された計算機システムは、相互に資源のアクセスを行なうための共通の機能を含むことにより、分散環境を提供することができる。ここでは、このような分散環境を形成するための、資源の名前空間に関する提案を行なう。

A Resource Managemt System for Large Distributed Computing Environmet

Jun MURAI
Computer Center,
University of Tokyo,
2-11-16 Yayoi, Bunkyo-ku,
Tokyo, 113, Japan.

and

Keisuke TANAKA
Dept. of Electorical and
Electoronnic Engineering,
Sophia University,
7-1 Kioi-cho, Chiyoda-ku,
Tokyo, 102, Japan

The progress of network technology provides distributed environent including resource access and sharing among widely interconnected computers.

Our resource management system treats any of resources as an '*object*', and defines transparent interfase to access these objects. And this system provides a name space for computer resources (*objects*) in the large distributed computing envieionment. All the resource are named in this hierarchical name space based on the domain system so that transparent specifications of resource names can be achieved.

1 はじめに

計算機ネットワークは、ARPA ネットワーク [1] をはじめとした、広域ネットワークの技術を起源として、ローカルエリアネットワークの普及を背景に、それらを接続するネットワーク間結合を基調としたインターネットワークのモデルへと遷移してきた。実際の計算機の利用環境としては、ローカルエリアネットワークにおける分散環境と、ネットワーク間接続を基盤にした広域資源アクセス環境との2つの環境に分類されている。前者は比較的高速のローカルエリアネットワーク技術を基盤として、透過性の高い環境を設定し、後者は広域ネットワーク技術を背景に、メッセージ交換やファイル転送などの限定された利用環境を設定している。また、これらの分類は実際には管理上の制限にも依存していて、前者は組織内の、後者は組織間の管理下におかれたそれぞれのサービスが提供されている。

このいずれの環境においても、利用している通信技術の進歩などの理由により、応用技術の発展は著しく、その結果として、インターネットワークのモデルにおけるこれらの2環境の明確な分離は困難になってきた。すなわち、この技術的状況は次のようにまとめることができる。

- 汎用のトランスポートプロトコルを用いた、ローカルエリアネットワークにおける局所的分散環境は、高度に発達している。
- 上記の環境を相互に接続する、広域通信は、十分に高速になり、技術的に局所的分散環境と同じプロトコルによるサービスを提供できるようになった。
- 局所的分散環境と広域的通信環境を分離する要素は、管理的側面であって、技術的な限界ではない。

ネットワーク環境における透過性の提供と、管理的な面や通信コストの面での資源へのアクセス制御とは独立に議論されるべき問題である。そこで、ここでは、広域接続を含んだ計算機環境全体に対応する広域分散環境の構築を実現するための透過的な資源名前空間とそれに基づいた資源管理機構について提案し、その設計についての議論を行なう。

2 技術的な背景

現在のネットワークによって提供される計算機環境は、広域ネットワークの技術を背景としてホストを要素とした、資源・名前管理の概念と、複数のオペレーティングシステムが一部の資源を透過的に共有することによる分散オペレーティングシステムとの概念とが共存することにより、さまざまな問題を生じている。この問題の背景を、DARPA ネットワーク体系用開発され、かつ、多くの分散オペレーティングシステム化への実験開発が行なわれているUNIX オペレーティングシステムを例にとり示す。

2.1 UNIX の名前管理

UNIX オペレーティングシステムの名前管理と資源管理は、一つのオペレーティングシステムを定義域とした階層的なファイルシステムによって主に実現されている。しかし、プロセ

スの名前はファイルシステムの名前空間に含まれていないために、プロセス間通信に用いられる名前に関する概念は一般に定義されていない。また、資源管理は、やはり一つのオペレーティングシステムを定義域として定義される利用者識別子によって行なわれている。UNIX ファイルシステム[2]では、各ファイルは、ディレクトリを定義域として一意に定まる名前が与えられる。各ディレクトリにもファイルとしての名前が定義されるので、全体では階層的な木構造を構成する。

2.2 UNIX ネットワークの名前管理

オペレーティングシステムが管理する資源は、一般的に一つのオペレーティングシステムを定義域とした名前空間によって行なわれている。したがって、ネットワークを介した資源へのアクセスは、一つのオペレーティングシステムが管理している空間の名前（ホスト名）と、そこで定義される資源の名前との組合せとして定義される。このホスト名を一意に定義する定義域は ARPA インターネットのホスト名の概念に従って、階層的に定義されるようになった[3,4] プロセス間通信機構においては通信の端点に名前がつけられ、ホスト内に限定したプロセス間通信には、ファイルシステムの空間で定義される名前が、ネットワークを介した通信には、DARPA プロトコル体系において定義されるサービスに対応する名前が使用されている。

2.3 分散ファイルシステム

NFS[5] や RFS[6] のような分散ファイルシステムでは、局所的オペレーティングシステムがそれぞれ管理するファイル空間を共有し、ファイルの名前空間で提供される資源の透過的なアクセスを可能にしている。UNIX の分散ファイルシステムでは、クライアントとなるホストから、サーバとなるホストのファイルシステムの一部を遠隔マウントして、クライアントのホストにおいてのみ、名前空間の拡張が行なわれる。したがって、環境全体で無矛盾性のある空間が提供されるわけではない。

2.4 資源・名前管理の問題点

ホストを定義域とした名前空間による分散資源のアクセスは、ホスト名と、そこで定義される資源名を用いて行なわれる。ここでは、分散資源の指定は各ホストで定義される名前の表現形式に依存して異なり、透過的な環境を提供することはできない。また、資源のアクセス管理は、各オペレーティングシステムで定義される利用者識別子によって行なわれるので、クライアントホストでの識別子とサーバホストでの識別子の対応を指定する必要がある。

分散ファイルシステムでも、この対応を陰に行なっていて、オペレーティングシステム単位の定義域、管理上の定義域、分散環境としてアクセスが可能な範囲の名前の定義域の3者がそれぞれ異なっている点に問題があることが分かる（表1）。

表 1. UNIX ネットワークにおける名前・資源管理

アプリケーション	名前表現
rlogin	<i>rlogin</i> [<i>HOST</i>]
	<i>rlogin</i> [<i>HOST</i>] -l [<i>USER</i>]
telnet	<i>telnet</i> [<i>HOST</i>]
rcp	<i>rcp</i> [<i>USER₁@HOST₁:FILE₁</i>] [<i>USER₂@HOST₂:FILE₂</i>]
	<i>rcp</i> [<i>USER@HOST:FILE</i>] [<i>LOCAL_FILE</i>]
mail	<i>mail</i> [<i>USER@HOST</i>]
talk	<i>talk</i> [<i>USER@HOST</i>] [<i>DEVICE</i>]

[*HOST*], [*USER*], [*FILE*](*[LOCAL_FILE]*), [*DEVICE*] には、それぞれホスト名、ユーザ名、ファイル名、デバイス名に対しての名前表現が割り当てられる。

3 広域分散環境における名前管理の概念

広域分散環境では、全体に対する統一的な名前空間を定義し、どのようなアーキテクチャに基づいて構築されたネットワーク内でも任意の資源へのアクセスの方法が透過的に提供されなければならない。そして、そこで定義された名前は、局所的に提供されている資源管理空間との対応付けが柔軟に提供される必要がある。この場合、それぞれの局所的な分散環境において定義されている管理上の制約は、広域分散環境上で定義される管理機構に対応することによって、実際的な管理面での要求を満足することができる。

3.1 名前の対応

広域分散環境が局所分散環境を提供するネットワークの集合を基盤として構成されることから、そのサービス提供のためには局所分散環境の提供する通信サービスが利用される。したがって、図 2. に示すように広域分散環境において定義される名前は局所的分散環境での名前に対応付けられる。この対応づけが的確でかつ十分な一般性を持った機能により実現されているのであれば、局所分散環境がいかなる概念のもとに構築されていても、その概念に依存することなく名前空間の構築が可能である。

3.2 名前空間の定義

分散環境では透過性の高い名前が実現されなければならない。また、管理すべき計算機資源が多数であるために名前の一意性の確保や名前情報の分散管理が容易でなければならない。この目的を達成するための一般的な方法として名前の定義域を階層的に設定するという方法が採られる。

全体の名前空間を組織などの管理上の境界に応じて幾つかの名前定義域(ドメイン)に分割し、それぞれに認識のためのラベルを与える、さらに各名前定義域をそこでの管理上の境界に応じてさらに小範囲を担当する名前定義域を設定する。この基本概念に従うことで全体

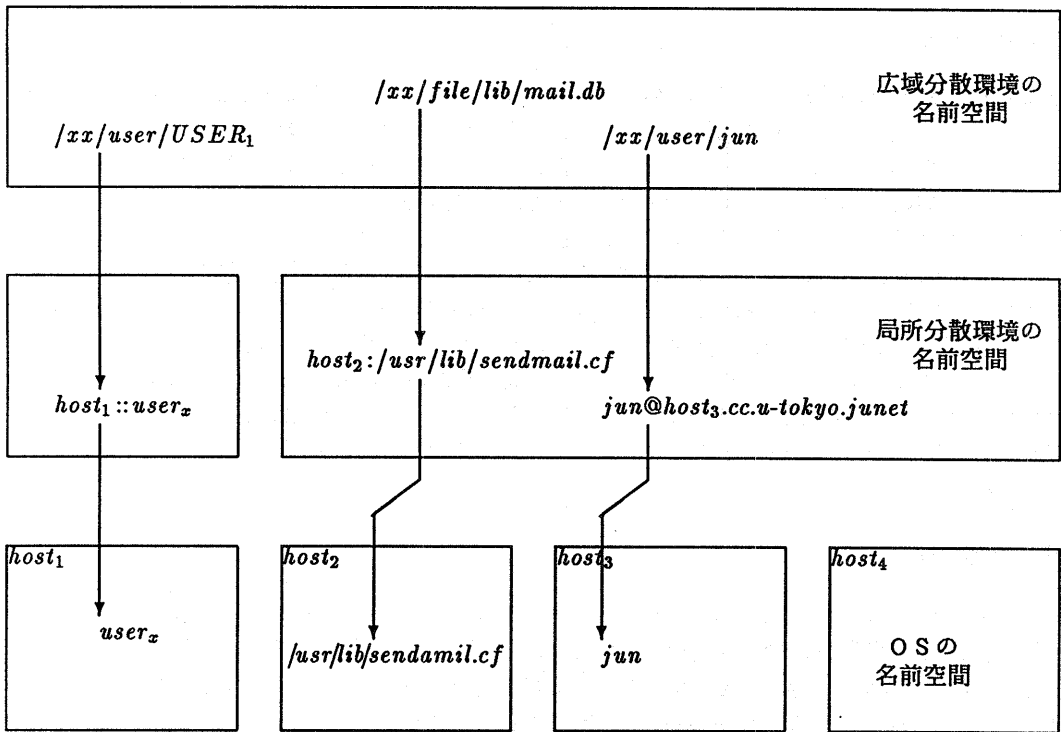


図 1. 広域分散環境の通信モデルにおける名前の関係

の名前空間は階層的に定義される名前定義域に分割される。この細分化された名前定義域がその定義域内での計算機資源の名前の一意性を保障し、その情報を責任持って管理すれば分散管理が可能である。

ここで、名前空間の分割は管理的側面にのみ基づくもので、物理的な要因は影響を与えていない。また、個々の局所分散環境やオペレーティングシステムに依存しない透過性の高い名前が提供される。さらに、管理上の境界を基準として名前空間が形成されるため、例えば、組織内の情報は該当組織内で管理責任を持つことになる。このため、この空間上での管理機構は容易であり、かつそれは実際の要求に沿ったものとなる。

3.3 資源管理空間

広域分散環境では全体に参加している個々のオペレーティングシステムには同質性を期待できず、したがって任意の資源を利用可能とするためには各資源は汎用の資源の概念で統合されなければならない。ここでは、ネットワーク上の存在するデータの集合である資源に対して、その存在を抽象化して取り扱うオブジェクトの概念を導入する。

各資源の利用はオブジェクトへの操作という形で一般化され、オペレーティングシステムに依存しない透過的な資源利用手段が提供されることとなる。個々の局所分散環境やオペレーティングシステムに応じた情報は全てオブジェクトの内部で管理され、対応付けられる。このことから、広域分散環境での資源管理はオブジェクトに対しての管理として定義される。

オブジェクトの実体はオペレーティングシステムや局所分散環境と密着した情報を管理するためその存在位置は物理的な制限を受けてしまう。このオブジェクトの位置を表す情報がオブジェクトのアドレスであり、全てのオブジェクトにはこのアドレスを用いることで通信が可能である。

4 名前管理機構

広域分散環境の利用がオブジェクトへの通信という形で定義され、個々のオブジェクトでその下層に存在する局所分散環境での資源管理機構との対応付けが行なわれるものとして、名前管理の主目的は論理的な名前空間上で規定される資源の名前(オブジェクト名)に対して、そのオブジェクトの実内容を保持する実体が存在する位置、すなわちアドレスを得ることである。オブジェクトのアドレスが得られれば、オブジェクトへの通信が可能となり、目的とする計算機資源の利用が可能となる。

ここでは、実験のための名前空間の構成を示し、この名前空間においてオブジェクト名からそのアドレスを得るための機構について述べる。

4.1 実験環境

名前管理機構の実験環境としてはまず、任意のホスト間での通信が可能でなければならない。しかし、全てのトランスポートプロトコルを統一することは事実上不可能であるため、これはネットワーク間のゲートウェイでのプロトコル変換という形で実現される。そして、このホスト間通信の上でオブジェクト間通信が可能となる。

そのためオブジェクトのアドレスは、全体のネットワーク内で一意に決定されるホストに対してのホストアドレスとそのホスト内で対象オブジェクトを限定するためのオブジェクトに対しての認識番号を利用して、以下の形式で表現される。

< *Global_Host_Address, Object_ID* >

このアドレスの形式は広域分散環境上で統一された形式で与えられるが、名前管理機構ではこれを単なるデータとしてしか取り扱わない。この情報の解釈は各々のオペレーティングシステム、局所分散環境がそれぞれの成立概念に応じた形で行なう。

4.2 名前空間の構造と名前表現

全体的な名前空間を幾つかの名前定義域(ドメイン)に分割する際、分割された個々のドメインには名前を表現するラベルが与えられる。個々のドメインの名前は階層的に上位のドメインによって管理され一意性を保障される。このようにすることでドメインとオブジェクトによる木構造をもった名前空間が形成される。ここで定義されたドメインはそれ自身もオブジェクトとして認識される。

同様の概念によって名前空間を構築する Clearinghouse[7] ではその使用対象がローカルエリアネットワークなど局所的分散環境であるためオブジェクトの識別のために三段階にドメインを設定した上、それぞれの階層の性格を規定している。しかし、広域分散環境では参加ホスト数は膨大な数にのぼり、管理対象となる資源数も限定出来ない、さらに、その規模も限定できない。したがって、ドメインの階層には制限を設けず、ドメイン設定の際の境界の種別も特定しない。

このようにして定義される名前空間においてオブジェクトは木構造の根から目的オブジェクトまでの経路上のドメインのラベルを '/' で結合した次のような書式で表わすこととする。

/domain₁/domain₂/.../domain_n/object_name

オブジェクトはドメインそのものを含めた広域分散環境下での管理対象となる全ての計算機資源を指す。すなわち、ドメインそのものが対象となった場合の名前表現ではドメイン名はオブジェクト名として定義される。

4.3 名前管理のための分散名前サーバ

名前管理を行なうための機構として各ドメインには名前サーバが存在する。個々の名前サーバは担当するドメインの名前管理情報を持ち、そのドメインについての名前管理の責任を負う。また、名前サーバ同志はオブジェクト間の通信機構を用いて相互に通信を行なうことが可能である。

初期状態で各名前サーバは木構造の上下のドメインの名前サーバに通信するためのアドレスしか保持していないが、名前解決のために得た他の名前サーバに対してのアドレス情報をキャッシュし、その後の名前解決作業の際にはこの情報を利用する。

名前解決の要求は任意名前サーバが受け付けることが可能である。名前解決の要求を受けた名前サーバは対象となるオブジェクト名を管理している名前サーバに論理的に最も近い

名前サーバを見つけだし、その名前サーバに対して通信を行ない解決を試みる。あるいは、適切な名前サーバが存在しない場合は木構造で上位に存在する名前サーバに対して通信を行ない解決を試みる。解決を依頼された名前サーバで未解決の場合は、その名前サーバがさらに適当な名前サーバを選択し、名前解決のための通信を行なう。このように再起的に名前解決が行なわれることで、いかなる名前に対しても、またいかなる名前サーバに解決要求が与えられても目的のオブジェクトのアドレスが得られる。

5 おわりに

広域分散環境では、個々の計算機システムで稼働するオペレーティングシステムの均一性を期待できないために、オペレーティングシステムの提供する環境や、そこに構築された局所的な分散環境に依存しない新しい資源・名前管理空間を形成する必要がある。そのためには、新たに設定された空間と、局所的な環境の空間とのマッピングが柔軟に行なわれる必要があり、また、処理効率に関する問題が解決される必要がある。

ここでは、このような概念に基づいた名前表現や名前管理機能の実証的な技術を確認するために設計した広域分散のための名前管理機構と資源管理機構について述べた。この機構を用いた名前表現の検討と、名前管理機構の効率の検証が今後の課題である。

References

- [1] V. G. Gref, E. Cain, 'The DoD internet architecture model,' Computer Networks, July 1983, pp. 307-318.
- [2] M. K. McKusick, W. Joy, S. Leffler, R. Fabry, 'A Fast File System for UNIX,' ACM TOCS, vol. 2, No. 3, August 1984, pp. 181-197.
- [3] P. Mockapetris, 'Domain Names - Concepts and Facilities,' RFC 882, USC/Information Science Institute, November 1983.
- [4] P. Mockapetris, 'Domain Names - Implementation and Specification,' RFC 883, USC/Information Science Institute, November 1983.
- [5] S. R. Kleiman, 'Vnodes: An Architecture for Multiple File System Types in Sun UNIX,' Proceedings of the USENIX Association Summer Conference, Atlanta, Georgia, June 1986, pp.238-247.
- [6] A. P. rifkin, M. P. Forbes, R. L. Hamilton, M. Sabrio, S. Shah, K. Yuah, 'RFS Architectural Overview,' Proceedings of the USENIX Association Summer Conference, Atlanta, Georgia, June 1986, pp.248-259.
- [7] D. C. Oppen, Y. K. Dalal, 'The Clearinghouse: A Decentralized Agent for Location Named Object in a Distributed Environment,' ACM Transactions on Office Information Systems, 1(3), July 1983.