

異種OS間連携を支援する拡張名前管理機能

高野 陽介 久保 秀士

日本電気(株) C&Cシステム研究所

NEBULA は、異なる種類のOSの間を橋渡しして、それらのOS上で実現されているサービスを1つの場所からOSの違いを意識せずに利用できるようにするシステムである。NEBULA が備えるサービスの名前管理機能は、複数のOSから集められたサービスの名前空間を利用者ごとの仮想的な名前空間に写像する方法を用いて、アクセス権の細かな設定、利用者ごとの便宜に適したサービスの名前空間の実現を可能にする。これまでに、この名前管理機能をUNIXシステムとACOS630システムの間で遠隔ファイル・サービスに適用したプロトタイプ・システムの実現の一部を完了し、その動作を確認した。

A Name Management Facility for Connecting Different Operating Systems

Yousuke TAKANO and Hidehito KUBO

C&C Systems Research Laboratories,
NEC Corporation

NEBULA connects different operating systems and enables to use remote services without knowing differences of operating systems. This paper describes the service name management facility of NEBULA, which provides an access control facility and service name space customizable for use, and a prototype system for remote file services among UNIX systems and an ACOS630 system.

1 まえがき

近年、コンピュータの利用範囲が広範になるのにとともに、コンピュータに対する利用者のニーズもますます多様化する傾向にある。これに対し、現状ではこのような利用者のニーズを1種類のコンピュータを用いて満足させることは難しく、用途に応じて異なる種類のコンピュータを使い分ける利用方法がしばしば見られる。

これを解決するためには、同種のコンピュータ間で行なわれているのと同様に、異なる種類のコンピュータの場合もネットワークで結合したシステムを構成し、その上に1つの場所から異なる種類のコンピュータのサービスをコンピュータの違いを意識せずに利用できるようにする環境を用意することが望ましい[4]。

このような観点から我々は、汎用機、専用機、WS等を含む異なる種類のコンピュータ群の上で遠隔サービスの利用環境を実現する試みをOS、および、ユーザインターフェイス[5]からのアプローチにより進めている。サービスとは、例えば計算能力、データ記憶の提供、データベース、プリンタの利用機能である。本稿では、このうちのOSからのアプローチとして構築を進めている複数OS間連携機構NEBULAについて述べる。NEBULAの目標は、異なる種類のOS、アウトOS、サーバを結合する橋渡しの部分を実現し、それらの間で計算機資源の相互アクセスを実現するための基盤となる機能を確立することである。現在、NEBULAのプロトタイプとして、UNIXシステムとACOS630システム間の遠隔ファイル・サービスを実現するシステムを試作中である。

本稿では、まず、複数OS間連携機構NEBULAの概要について述べる。次に、NEBULAの提供する機能の1つであり、異種OSの保有する計算機資源等の名前、アクセス権の操作に関する連携を実現する名前管理機能について述べ、最後に試作の一部が完了しているプロトタイプ・システムについて述べる。

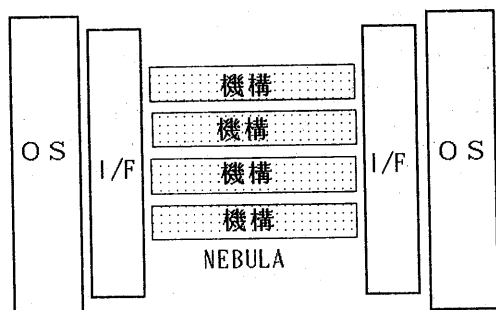
2 複数OS間連携機構の構成

複数OS連携機構NEBULAの設計方針の1つは、複数種のOSにわたるサービスの相互利用環境を構築するための部品群を供給することである。これにより次のような効果があると考えられる。

- 多様なコンピュータの結合形態、多様な利用者の要求に柔軟に対応するシステムを構築することが可能になる。様々な種類のコンピュータ、OS、サービスの組合せに対して定型的なシステムで対応するのは難しい。そこで、結合を支援するための標準的なツールや部品群を用意し、状況ごとに必要な部分を実現する方が個々の状況における要求に柔軟に対応できると考えられる。
- 特定の機能を特定の応用に向けて最適化したいなどの要求に対して、特定の部品に注目して改善することができる。例えば、ファイルの転送速度を向上させたい場合には、ファイル・アクセスを実現する部品に注目する。
- 既存のコンピュータ連携機能の機能向上に利用することができる。コンピュータ間に既存の連携機能が用意されている場合に、それを改造してNEBULAの部品に充てることが可能になる。例えば、既存の遠隔ファイル・アクセス機能をNEBULAの部品に改造して、NEBULAが用意する名前管理の部品をそれに連結すれば、NEBULAの名前管理機能を付加されたファイル・アクセスの環境が構築できる。

NEBULAの部品は、図1のように、複数のOSの間を連絡する橋渡しの機構であり、実際には接続されるコンピュータの両側に処理プログラムが配置される。

NEBULAは、個々のOSとは基本的に独立で標準的な機能仕様を実現する。対象マシンへ接続する際には、接続先に依存した機能を備えるインターフェイス部分を作成し、それを經由してOS



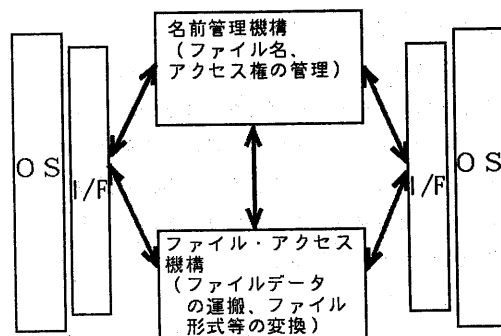
I/Fはインタフェース部分

図 1: NEBULA の構成

に接続する。このため、NEBULA の機能仕様の多くはインターフェイス部分を実現するプログラムに対するものである。NEBULA はリモート・マシン群から提供されるサービスを標準的な仕様に取りまとめる役割を持っており、その標準仕様をローカル・マシンにどのようなセマンティックスで見せるかはインターフェイス部分の作り方に依存している。例えば、後述のプロトタイプ・システムでは、NEBULA が取りまとめた複数の OS からのファイルを、インターフェイス部分が UNIX のファイルとほぼ同等の仕様で見せるように変換している。インターフェイス部に関しては 4.2 節で、プロトタイプ・システムにおける UNIX カーネルへのインターフェイスの事例を引用して説明する。

NEBULA の部品を大きく分けると、サービスの種類ごとに存在するサービスへのアクセス機構、および、それらの機構に共通に使われる基本機能を提供する共通機構の 2 つの機構が存在する。

サービスの種類ごとに存在するアクセス機構としては、ファイル・アクセス機構、サーバ・アクセス機構、トランザクション機構などを想定している。また、共通機構に関しては、名前管理機構、課金管理機構、利用者管理機構などを想定している。現在、設計・開発が開始しているのはこのうちのファイル・アクセス機構と名前管理機構



I/Fはインタフェース部分

図 2: 名前管理機構とファイル・アクセス機構の関係

の組合せである。この 2 つの機構の役割はおおよそ図 2 のような分担になっている。

名前管理機構は、サービスの名前とアクセス権を管理する役割を果たす。例えば、ファイル・アクセス機構に対して機能する場合には、ファイルの名前とアクセス権を管理する。

ファイル・アクセス機構はファイル・データの運搬、キャッシング、データ形式、ファイル形式などの変換を行なう。OS 間でファイルを受け渡す際に、ファイルを持っているサイトを確認したり、ファイルのアクセスが許諾されるかを確認するために名前管理機構と情報を交換する。

次の節では、NEBULA で設計した名前管理機能の特徴について述べる。

3 名前管理機能の設計

NEBULA の名前管理機能は複数の OS から集められたサービスに対する名前空間を実現する。名前制御の主な目的は次の 2 点である。

1 つは、アクセス権制御の強化である。本システムの主な適用範囲として、汎用機、専用機などの比較的高価な、また機密性の高い計算機資源を保有するコンピュータと WS などの結合を前提としている。このため、名前管理機能の設計に当たっては、資源アクセスに際してより細かくアク

セス権を制御でき、また課金のための情報を収集できる能力を必要としている。

もう1つは、サービスの存在、所在、利用を開始するための手続きを利用者からより分かり易くすることである。特定の仕事を遂行するために必要ないくつかのサービスを、それを実際に提供しているOSとは独立に1まとまりに集めておき、簡単な手順でそれを呼び出せるようにすることを目指している。

分散ファイル・システムなどの実現で用いられている名前管理の方法を上記の目的でサービスの管理に適用する場合いくつかの不足を生ずる。既存の分散ファイル・システムでは、OS間でのファイル名前空間の共有に関して大きく分けて2つの方法が使用されていると考えられる。1つはAFS[3]やVシステム[1]などで用いられている方法で、サイト間で1つのグローバルなファイル名前空間を共有するものである。もう1つは、NFS[6]やSAA[2]などで用いられている方法で、共有可能なファイル名前空間をサイト間で個別に契約するものである。前者の方法では、高価なサービスに対する細かなアクセス権の設定が難しい。後者の方法では、複数のサイトから名前を集める際の自由度が制約され、使い易い名前空間が構成できない可能性がある。

以下では、このような不足を補うことを目的としたサービスの名前空間の機能について説明する。

(1) 基本的な性質

NEBULAは、図3に示すような実名前空間と仮想名前空間の2種類の空間を実現する。実名前空間はサービスを管理するために用いる空間である。複数のOSにわたるグローバルな空間で1つだけ存在し、複数のOSからNEBULAに集められたすべてのサービスの名前を写像する。サービスは、OSからNEBULAに提供された際の名前・アクセス権で写像されている。また、実名前空間の内部の構造はサービスの管理の便宜に基づいて構成する。例えば、提供元のOS毎やサービスの種類の分類で整理する。

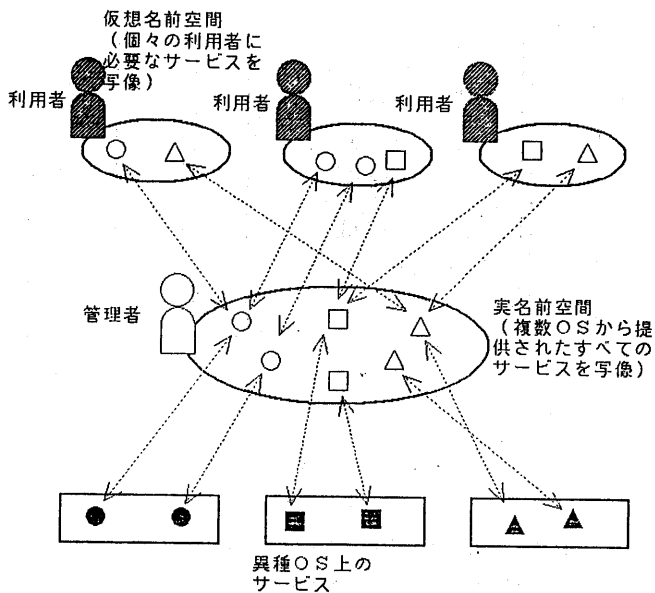


図3: サービス名前空間の実現

これに対し、仮想名前空間はサービスを利用するために用いる空間である。実名前空間の一部を写像することによって構成され、利用者、用途などに応じて複数個定義できる。実空間から写像する際に、サービスの名前は別のものに置き換えることが可能で、アクセス権は縮小する方向での変更が可能である。仮想名前空間の内部の構成は、サービスの利用から見た便宜に基づいて構成することができる。このように、実名前空間と仮想名前空間の設定は、管理および複数の利用者に対して名前空間の構成を独立させ、それぞれの使い易さを向上させることができると考えられる。

(2) 名前空間の所有者

NEBULAでは、すべての名前空間にその所有者を定義している。実名前空間の所有者はNEBULAの管理者である。仮想名前空間の所有者は例えば、特定のサイト上の利用者、利用グループである。所有者の概念と個々のOS上の利用者の概念の間の対応付けはOSへのインターフェイス部分の設計に依存している。

所有者の原則として、名前空間の所有者のみがその名前空間を用いてサービスにアクセスできる

ことにしている。すなわち、名前空間はその所有者からの要求に対してのみ可視になる。さらに、名前空間ごとにサービスのアクセス権が定義されるため、所有者ごとに細かくアクセス権が定義できることになる。これらにより、名前空間という枠を用いて利用者が利用できるサービスを制限し、高価なサービスが濫用されるのを防止しようとしている。

所有者の設定は、名前管理機能がサービスの所有者をどのようにして識別するかという実現上の問題を含んでいる。後述するプロトタイプ・システムでは、ファイル・アクセスの際に UNIX の実効ユーザ ID と実効グループ ID が UNIX カーネルから名前管理機能に直接渡され、それが参照して名前管理機能は特定の利用者を認識し、ファイルの名前空間の可視性を決定している。この他に、所有権をケイバリティとして実現し、それを用いて NEBULA が正当な所有者を認識する方法も可能であると考えられる。

(3) 名前空間に対する管理

NEBULA の管理者は、サービスの実名前空間から仮想名前空間を生成し、NEBULA 上のサービスを利用したい利用者にそれを譲渡する役割を果たす必要がある。すなわち、実名前空間から仮想名前空間へのサービスの写像、アクセス権の変更、所有者の設定である。しかしながら、この状態では管理者にかかる負担が大きいため、管理者の役割を代行する支援機構の実現が不可欠である。

名前管理の支援機構が果たすべき役割として必要と考えられるのは次の機能である。

- 典型的な仕事に対応した名前空間の鋳型の供給とそのインスタンスの生成
- サービスとその性質・特徴を記録したデータベースの保持と検索手段
- 利用者の分類と分類毎に与えるアクセス権のガイドラインの運用

(4) 名前空間の利用

利用者は、所有権を持っている仮想名前空間をローカルなオペレーティング・システムの名前空間に接続して利用することができる。名前空間をローカルな名前空間に写像する方法に関しては、やはり NEBULA とローカルなオペレーティング・システムの間インターフェイス部分の実現方法に依存している。

4 プロトタイプ・システムの実現

NEBULA のファイル・アクセス機構と名前管理機構のプロトタイプ・システムを UNIX マシン (EWS4800) と汎用機 (ACOS630) をイーサネットで結合したハードウェア上に適用する作業を進めている。名前管理の機能は前の節で述べた仕様のサブセットになっている。ここでの目的は、サービス管理方式の試作と性能の評価を行なうこと、および、今後本プロジェクトに必要とされる複数の OS にわたるプログラム開発環境を整備することである。現在、UNIX のファイル・システム上から ACOS630 の一部の種類のファイルをシステム・コール・レベルでアクセスできる環境が完成している。

4.1 プロトタイプ・システムの構成

プロトタイプ・システムの構成を図4に示す。2台の UNIX システムと1台の ACOS630 システムがあり、1台の UNIX システムをファイルの利用サイトに割り当て、もう1台の UNIX システムと ACOS630 システムをファイルの供給サイトに割り当てている。ファイル・アクセス機能、名前管理機能は UNIX と ACOS4 の両側に分散して配置している。UNIX、ACOS4 間の通信は、OSI セッション層通信機能を使用している。利用サイトの UNIX 上の名前管理機構とファイル・アクセス機構は独立したサーバ・プロセスによって実行している。さらに、利用サイトの UNIX 上のファイル・アクセス機構は UNIX インターフェイス機能を経由して、UNIX のカーネル内部の vnode インターフェイスに接続して

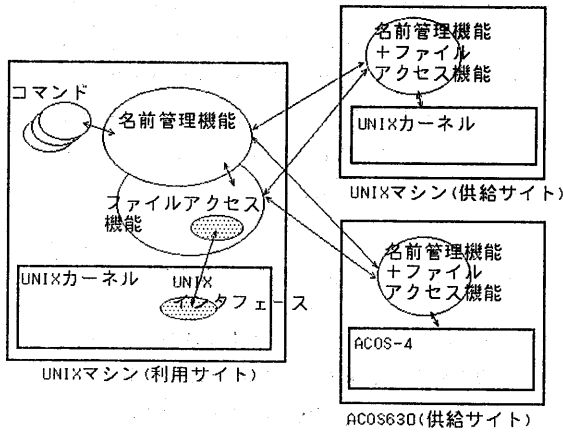


図 4: プロトタイプ・システムの構成

いる。UNIX インターフェイス機能は、UNIX カーネル内部とファイル・アクセス機能のサーバ・プロセスに含まれる。供給サイトの ACOS630 と UNIX 上の機構は 1 つのプロセス (ACOS630 ではジョブ) で実現している。個々の機能の役割は次の通りである。

UNIX インターフェイス機能は、NEBULA の提供するファイル名、アクセス権、ファイル・データへのオペレーションと UNIX 内部で発生するオペレーションの間の構文、意味の変換機能、および、NEBULA と UNIX カーネルの間の通信機能を含む。

ファイル・アクセス機能は、UNIX カーネルから UNIX インターフェイスを経由して要求を受け付け、供給サイト上のファイル・アクセス機能と通信を行なってファイル・データの受け渡しを行なう。また、ファイル名やアクセス権の操作の要求の際には名前管理機能に通信を行ない必要な情報を取得する。

名前管理機能は、供給サイトの OS から提示されたサービスをいったん収集し、アクセス権の付加と名前の操作を行なったのちにファイル・アク

セス機能に提供している。このため、内部にシステム管理者が定義した仮想名前空間の情報を保持しており、仮想名前空間の中ではアクセス権、実名前空間へのポインタがファイル名 1 つ 1 つに対して定義されている。さらに、階層ディレクトリ、および、その上でのリンク、シンボリックリンクの情報を保持でき、個々のファイル名はその上に配置できる。ファイル・アクセス機能から、ファイル名検索、ファイル名生成・変更・抹消、ファイル状態取得・変更、ディレクトリ/リンク/シンボリック・リンクの生成・抹消の命令を受け付ける。さらに、管理者用の名前管理コマンド・プログラムからの要求を受け付ける別の窓口を持ち、ファイル名前空間の作成・検索・抹消、ファイル名前空間への名前の写像などの要求を受け付ける。コマンド・プログラムは、名前管理機能の指定するポートに対して要求番号と引数を送信することにより要求を伝達している。

例として、UNIX 上の利用者プロセスから ACOS4 上に実体のあるファイルの書き込みが発生した場合の処理経過について説明する。書き込み要求はまず UNIX カーネル内の vnode インターフェイスによって、ローカルな UNIX ファイルか NEBULA の名前空間上のファイルのいずれへの書き込みであるかが判定される。NEBULA のファイルの場合、UNIX インターフェイスによって、その要求は書き込みデータと共にファイル・アクセス機能に伝達される。ファイル・アクセス機能では、書き込み対象が ACOS4 ファイルと UNIX ファイルのいずれかであるかが判定される。ACOS4 ファイルの場合、要求と書き込みデータは ACOS4 上のファイル・アクセス機能に渡され、そこで ACOS4 のファイルの実体へ書き込まれる。

4.2 UNIX インターフェイス

プロトタイプ・システムでは、UNIX インターフェイス機能により、UNIX 上の利用者プログラムの大部分から、UNIX のシステム・コールを通して、ACOS4 ファイルが UNIX ファイルと

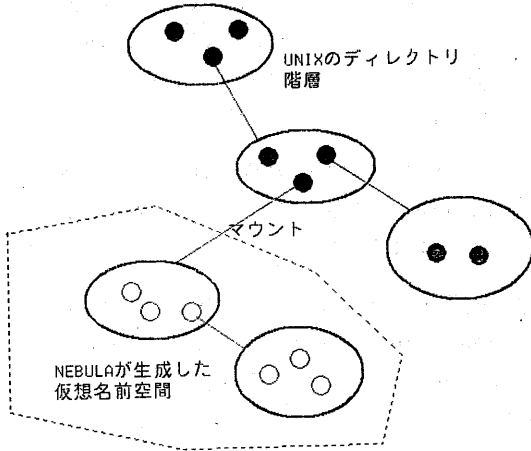


図 5: UNIX からの名前空間の見え方

ほぼ同等に操作できる環境を実現している。

まず、管理の面から見た機能に関して説明する。これまでに、NEBULA のファイル名前空間の管理を行なうため、以下のコマンドを実現している。

ファイル名前空間の生成は、次のコマンドで行なう。

nsgen 名前空間名 利用者名

これにより、空の名前空間が1つ生成される。作成された名前空間は、コマンドの引数で指定した利用者からのみの参照が可能ないように設定される。利用者名の代わりにグループ名を指定した場合には、そのグループに属する利用者からのみの参照が可能になる。

サーバから提供されている既存のファイルを、ファイル名前空間の上に別の名前、別のアクセス権で写像するためには次のコマンドを発行する。

nsmap サーバ名:ファイル名

名前空間名:パス名 アクセス権

このほか、ファイル名前空間の一覧の参照、ファイル名前空間の抹消の機能を有するコマンドを実現している。

次に利用の面から見た機能に関して説明する。UNIX インターフェイス機能は、名前管理機能が実現するファイル名前空間を SUN の NFS に近

い方法で UNIX のファイル・システムに接続する。接続は図5のように UNIX のファイル階層上のディレクトリに名前管理機能が実現するファイル名前空間を次のコマンドによりマウントする。

nsmount 名前空間名 ディレクトリ名
 名前空間名は、nsgen コマンドを用いてシステム管理者が付けた名前である。これによって、指定したディレクトリ配下には供給サイトである ACOS4 と UNIX のファイルが混在した名前空間が接続される。

マウントによって接続された名前空間を参照できるのは、ファイル名前空間の生成の際に指定された利用者のみである。それ以外の利用者がマウントしたディレクトリの配下を参照しても、マウントされていない状態が参照される。このように、UNIX から見て、他の OS から供給されたファイルを含むファイル名前空間は特定の利用者によって排他的に利用できるようになっており、ACOS4 のファイルは指定された特定の利用者以外から保護された状態で利用されている。

ファイル名前空間の内部には、UNIX のファイル名前空間と同様に、ディレクトリ、リンク、シンボリック・リンクが作成できる。また、利用者がファイル名前空間上のファイルを新規生成する場合、原則としてファイルの実体は名前空間毎に設定されるデフォルトのサイト上に割り付けられる。UNIX 上の名前管理機能は、このデフォルトのサイトを指定する窓口を利用者に対して提供しており、利用者はサーバの指定するポートに通信することでこの設定を変更できる。

現時点の機能では、利用者の使用するファイル名前空間はシステム管理者によって生成されるため、システム管理者にかかる負担が多くなっている。すなわち、利用者はサービスの利用を開始する時に利用者だけの手続きではこれを行なえず、手続きはシステム管理者に委ねられることになる。これについては、前述のような名前管理を支援するツールを実現することによって対応する予定である。

5 むすび

異種OS間の相互資源利用環境の実現を支援するNEBULAの概要、NEBULAの部品の1つである名前管理機能、そのプロトタイプ・システムに関して述べた。NEBULAの実用を可能にするまでには解決しなければならない問題が多い。名前管理機能に関しては、保守性、セキュリティなどの問題が解決されていない。またファイル・アクセス機能に関しては、様々な種類のファイル編成やデータ形式に対応する手段が用意されていない。今後の課題は、これらの問題解決のための手法を検討・実現すること、および、利用者管理機能などのそのほかの共通機能に関して検討を進めることである。

謝辞

本研究の機会を与えていただくと共に、有益なご指導、ご助言を与えて下さった日本電気(株)石黒辰雄支配人、C&Cシステム研究所の小池誠彦部長、中崎良成課長、難波信治主任をはじめとする方々に深く感謝いたします。また、プロトタイプ・システムの実現に携わっていただいた日本システムウェア(株)の平野守氏をはじめとするプロジェクトチームの方々に深く感謝いたします。

参考文献

- [1] D. R. Cheriton. The v distributed system. *Communications of the ACM*, 31(3):315-333, 1988.
- [2] R. A. Demers. Distributed file for saa. *Ibm Systems Journal*, 27(3):348-361, 1988.
- [3] J. H. Howard. An overview of the andrew file system. In *Proceedings of Usenix Conference*, pages 23-26, 1988. Winter.
- [4] D. Notkin, N. Hutchinson, J. Sanislo, and M. Schwartz. Heterogeneous computing environments, report on the acm sigops workshop on accommodating heterogeneity. *Communications of the ACM*, 30(2):132-140, 1987.
- [5] 岩崎未知. 異機種分散環境における uims の構成. 情報処理学会第 41 回全国大会講演論文集 (5), 5-221, 1990.
- [6] D. Walsh, B. Lyon, et al. Overview of the sun network filesystem. In *Proceedings of Usenix Conference*, 1985. Winter.