

稠密処理システム GALAXY での
ネーミング

3U-9

濱野 純、前川 守、太田昌孝、河内谷清久仁
(東京大学)

1. はじめに

現在、われわれは、広域ネットワーク上で多数のワークステーションが有機的に協調動作する処理形態の研究を進めており、そうした処理形態を稠密処理 (Holonc Processing) と呼ぶことにしている。われわれが稠密処理の概念に基いて設計しているコンピュータ・システムは GALAXY と呼ばれている。

一般に、ネットワークによって接続されたコンピュータ・システムでは、ユーザからアクセス可能な (少なくとも、物理的には接続している) 計算機資源の種類や数は、スタンドアロン・システムの場合に比べて桁ちがいに大きくなるが、GALAXY の場合もまたその例外ではない。

こうした状況では、それぞれのシステム構成要素をユーザからどう呼ばせるか、という、ネーミング方式に関する考察が重要なものとなる。今回の発表では、われわれが GALAXY でとったネーミング方式と、それを決定する際に考慮したことがらのいくつかについて議論する。

2. GALAXY のネーミング方式に対する要求

GALAXY のような分散化環境においてユーザがシステム構成要素を名指しする場合、その名指しの方法はおよそ以下のような性質を持っていることが望ましいと考えられる。

- 1) システム構成要素 (以下では GALAXY の用語に従って、オブジェクトと呼ぶ) の名前と、オブジェクトの内容 (GALAXY の用語では、エンティティと呼ぶ) が存在する物理的な位置 (ロケーション) とが互いに独立であること。

この性質が実現されると、GALAXY 内のすべての情報及び資源を、そのロケーションによらず統一された形で名指すことができるようになる。

- 2) 同一のオブジェクトを指す名前は、それを名指すユーザやプログラムのロケーションによらずに決まること。

この性質が実現されると、ユーザはどの位置にいても、全く同じ GALAXY システムを見ることが

できる。

これらは、一般に、ロケーション・トランスペアレンシ (location transparency) と言われる性質である。

すでに Unix 等の上にネットワーク・ワイドな名前空間を構築しようとした例は存在する [1] が、それらはリモート・ファイルシステムをマウントすることを基本としたもので、ロケーション・トランスペアレンシを実現していない。完全にロケーション・トランスペアレントなシステムを構成しようというのが GALAXY の一つの目的となっている。

- 3) さまざまな種類のオブジェクトが同一の名前空間内に置かれること。

この性質が実現されれば、各種オブジェクトの統一的な管理が可能となる。

Unix はファイル、ディレクトリとデバイスを同一のファイル・システム内に実現して成功をおさめているが、GALAXY では、それを一歩すすめて、ユーザや部分システム、OS 自身もまたオブジェクトの一つとして同一の名前空間に置こうとしている。このことによって、名前空間での探索 (Unix では find) コマンド一つで、ファイル名もユーザ名も同一の手順で探索できるようにしようというのがそのねらいの一つである。

- 4) エンティティとオブジェクトの名前とは独立であること。

この性質が実現されれば、オブジェクトを使用している最中にその名前だけを変更することができ、なおかつ、オブジェクトを使い続けることができる。

- 5) 名前空間が階層的な構造を持っていること。

この性質が実現されれば、相互に関連のあるオブジェクトを他のオブジェクトと区別し、ひとまとめにして扱うことが可能となる。

- 6) ユーザ毎に、自分が使いやすいと思う名前の書き方や短縮形、省略形を用いることができること。

この性質が実現されれば、ユーザは自分向きに快適な操作環境を得ることができる。

CMU の Andrew システムではワークステーション毎にネットワーク上のファイル・システムとワークステーション側の OS (普通は Unix) との名前付け法の変換システムを持つことで各種 OS を用いたワークステーショ

Naming of GALAXY Holonc Processing System

Jun Hamano, Mamoru Maekawa, Masataka Ohta, Kiyokuni Kawachiya
University of Tokyo

ンの参加を可能としている [2] が、GALAXYではワークステーションではなく、ユーザ毎に名前付け法が選べるようにした。

3. GALAXYの3レベル・ネーミング方式

上であげた各要求を満たすために、GALAXYでは3レベルの名前付けを行なうことにした。各レベルは、

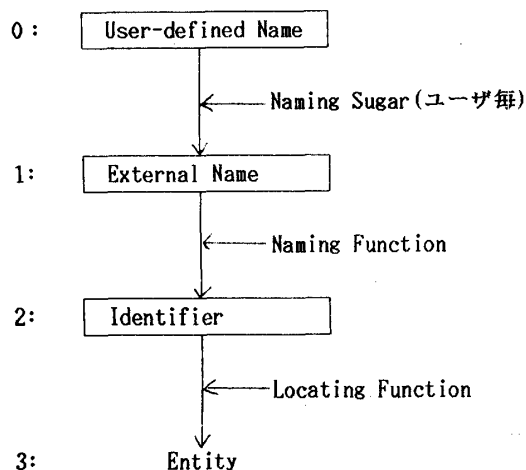
- 1) External Name レベル
- 2) Identifier レベル
- 3) Entity レベル

と呼ばれている。その外側にさらに、

- 0) User-defined Name レベル

が存在する。

この様子を図に示す。



最外層である User-defined Name 層によって要求 6) が実現される。この層は名前付け機構とユーザとの高級インタフェースであり、この層での名前はテキスト名である。User-defined Name は、ユーザ毎に指定され、ユーザ・モードで (すなわち、オペレーティングシステムの一部としてではなく) 実行される Naming Sugar によって、次の External Name 層の名前に変換される。

この層によって、Unix 風のパス名の書き方、たとえば相対パス名表記や文字～によるホーム・ディレクトリの表示などをシミュレートすることができる。

Naming Sugar は名前付け機構から起動されるために、ユーザが陽に指示したオブジェクト名だけでなく、ユーザにより起動されたプログラムからのオブジェクト参照も全てその恩恵に浴することができる。これは便利であるが、保安上有害な場合もありうるため、Naming Sugar をバイパスして、直接に External Name を記述することも許すことにしている。

External Name 層は、要求 4)、5)を実現するための層である。この層での名前はテキスト名であり、GALAXYシステム内の全てのオブジェクトを含む巨大な一本

の木構造をなしている。

GALAXYのオブジェクトの種類の一つに、ディレクトリ・オブジェクトがある。ディレクトリ・オブジェクトはそのディレクトリのメンバであるオブジェクトの External Name と、一つ下の層である Identifier との組を保持している。GALAXYの名前付け機構の一部である Naming Function が、この情報を用いることによって External Name を Identifier に変換する。

通常、ユーザが見るオブジェクトの名前は External Name である。オブジェクトを使用する際には、まず最初にその Identifier を External Name から得て、その Identifier によってアクセスするという手順を踏むため使用中のオブジェクトの名前を変更しても問題は起らない。

後に述べるように、ロケーション・トランスペアレンシは Identifier 層で実現されるために、External Name の木構造は純粹に論理上の理由のみで構成されるものとなり、「ここから下の部分木はどのワークステーションが持っている」といった、リモート・ファイルシステムのマウントによる方式とは本質的に異なる利点を持っている。

Identifier 層は、要求 1)、2)、3)を実現するものである。GALAXYはオブジェクト生成時に、オブジェクトと1対1に Identifier を与える。この Identifier はシステム全体で一意的なものであるが、それ自身 (バイト列としての Identifier の値) にはロケーションの情報等、一切の情報を持たない (Unix において i-number の値そのものに特別な意味がないのと同じである)。GALAXYの各ワークステーションには ID表と呼ばれる表があって、各ワークステーションが必要とするだけの Identifier について、それに対応するオブジェクトに関する物理的情報が記録されている (これを Entity Information と呼んでいる)。GALAXYの名前付け機構の一部である Locating Function が、この情報を用いることによって Identifier から Entity の位置を決定し、オブジェクトへの実際のアクセスが行なわれる。

4. まとめ

今回の発表では、稠密処理システム GALAXY とユーザとのインタフェースであるネーミング方式について述べ、その考え方と実現法について議論した。

References

1. Luderer, G.W.R. et al., "A Distributed Unix System Based on Virtual Circuit Switch," in Proc. 8th Symp. Oper. Syst. Princip., pp. 160-168, 1981.
2. Morris, J.H. et al., "Andrew: A Distributed Personal Computing Environment," CACM, vol. 29, 3, pp. 184-201, 1986.