

6V-3

分散型OSにおける
ネットワーク管理の一方式

計 宇生 斉藤忠夫 猪瀬 博

東京大学工学部

1 はじめに

急速に発展しているコンピュータ・ネットワーク技術の分野で、近年、ローカル・エリア・ネットワークを通信媒体とし、資源の共有及び有効利用を目的としたローカル・コンピュータ・ネットワークに関する研究は特に盛んである。本稿では、当研究室で構築されている分散型オペレーティング・システムCOSMOSを対象とした場合のネットワーク管理機能について述べる。

2 COSMOSの概要とネットワーク管理の役割

COSMOSは、LANで結合された計算機システムのOSであり、UNIXの基本構想をベースにし、ネットワーク環境においてハードウェア、ソフトウェア資源の共有及び負荷の分散機能を実現することを目的としている。図1にCOSMOSを搭載したシステムの論理構成を示す。各種の機能を論理的に別ノードに分散させることが特徴である。

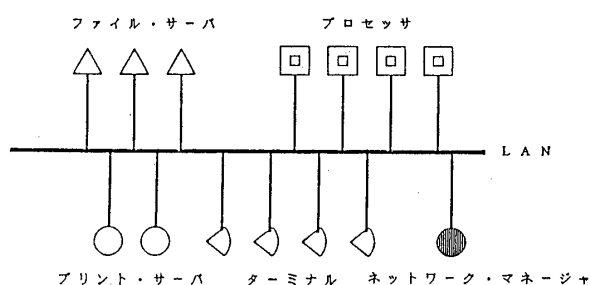


図1 COSMOSシステムのノード構成

このようなシステムにネットワーク管理をOSの一部として導入することによって、ネットワーク透明な分散処理環境をユーザに提供する。ユーザが出したネットワーク資源に対するアクセス要求に対して、ネットワーク管理プログラムは各ノード上のカーネルと協調して完成し、ユーザにネットワークを意識させない。またCOSMOSにネットワーク均質性を持たせる。各ノードの構成及び機能が違う不均質なネットワークにおいて、ユーザがどこのノードからネットワークを見渡しても同じようなイメージを持つ。COSMOSではネットワーク管理部分が各ノードの共有資源に統一した名前を与え、資源の局所性をなくすことによってネットワーク均質性を

提供する。そして、COSMOSで実現されるもう一つの機能はシステムによる処理負荷の分散である。ネットワーク内の処理負荷が一ヶ所に集中した場合、それらを適当に各ノードに分散させることによってシステム全体のスループットを上げる。

以下ではネットワーク管理の実行の主体をネットワーク・マネージャと呼び、COSMOSではネットワーク・マネージャによる集中的ネットワーク管理を考える。

3 総合的なネットワーク管理機能

COSMOSでは、ネットワーク管理プログラムを用いてネットワーク内のすべてのノードを統合的に管理する。ネットワークの正常動作時に、各ノードの生存情報やノードの属性などのような情報を管理する。それらの情報はノードとしての資源を利用する時に常に必要である。異常が発生した場合ユーザ・プロセスへ通知し、それに応じた適当な事後処理を行うことができる。

COSMOSのノードの管理はネットワーク・マネージャがノード管理テーブルを用いて各ノードの活動状況を定期的にモニタすることによって実現される。ユーザの処理の中でネットワーク内のノードに関する情報は必要な時にネットワーク・マネージャから情報を得ることができる。また、ノードの配置などの変更をするとき、ユーザがシステムを通して行うことができる。

以上の他、ネットワーク・マネージャはさらに時刻の管理やユーザ管理の機能を勤める。

4 オブジェクト間通信機能

COSMOSではネットワーク内の各種の資源、プロセス、ファイル、デバイス、パイプなどをまとめてオブジェクトという概念で統一的にとらえる。各オブジェクトはネットワーク内で一意的に決まるオブジェクト識別子(oid)を持っており、オブジェクトに対する操作及び管理はoidを用いて行う。ユーザからはoidが見えない。

COSMOSにおけるオブジェクト間通信の種類は主に(1)プロセス型→プロセス型、(2)プロセス型→ファイル型である。(1)プロセス型オブジェクトは、そのオブジェクト識別子oidから場所の指定ができるはずである。しかし親プロセスが子プロセスと通信したい場合、

The Network Management for a Distributed OS

Yusheng JI, Tadao SAITO, Hiroshi INOSE

University of Tokyo

親プロセスによって生成された子プロセスは実行の場所が負荷分散のため別のノードに移動される場合がある。その存在場所はプロセスの割り当てをするネットワーク管理プロセスのみ知っている。従って、親プロセスと子プロセス間の通信はネットワーク・マネージャを通ることになる。(2)COSMOSでは、パスネームに従って目的ファイルを探し出すまでの間に、複数のファイル・サーバ間を渡り歩く場合もありうる。この時の通信のトランザクションは一般にネストした通信となる。しかしトランザクションがネストしていると、ファイル・サーバ上のサーバ・プロセスが一つのユーザ・プロセスからの要求に対してサービスする時間が長く延びてしまい、処理の効率が悪くなる。途中で何かの障害が起きた場合ユーザ・プロセスとは直接の通信ではないため回復も困難である。従ってCOSMOSではポイント・ツー・ポイントの通信の方法を採用し、一旦ユーザ・プロセスを経由してから他のファイル・サーバと通信する。

5 ネットワークワイドなプロセス管理

COSMOSでは、ネットワーク・マネージャがネットワークワイドなプロセスの管理を行う。しかし、ユーザ・プロセスの実行はすべてプロセッサ・ノードで行なわれ、ネットワーク・マネージャはユーザ・プロセスの実行を一切行なわない。そのため、ネットワークワイドなプロセスの管理以外に各プロセッサに割り当てられたプロセスに対して、そのプロセッサにおけるローカルな管理が必要である。そこで、それらを二つのレベルでの管理と見なし、前者を高次的なプロセス管理、後者を低次的なプロセス管理と呼んでいる。

5.1 高次的なプロセス管理

ネットワーク・マネージャはプロセッサの管理テーブルを用いて、プロセッサについての情報を管理する。それらの情報の中で、負荷平均やメモリーの使用量は常に変動している量であるので、それらを把握するためには各プロセッサからネットワーク・マネージャに報告する必要がある。また、プロセスをプロセッサに割り当てる時に、処理の内容にも配慮しなければならない。その処理がCPUを主に必要とするCPUバウンドの処理であるか、それとも端末との入出力が中心となるI/Oバウンドの処理であるかを区別して、後者の場合はできるだけ他のノードへ割り当てないようにする。

5.2 プロセスの生存過程

COSMOSのプロセスの生成、実行及び消滅は次のように実現される。

- ・プロセスの生成: forkシステム・コールによって一つの新しいプロセスが生成される。この時新しいプロセスにプロセス識別子を与えるので、プロセスの統合的管理者であるネットワーク・マネージャへ通知する。

- ・プロセスの実行: ネットワーク・マネージャは手持

ちの各プロセッサの生存、使用状況などの情報に基づいて、実行するプロセッサを指定する。このことはシステム・コールexecが発生した時に行なわれる。

- ・プロセスの消滅: プロセスが生成(fork)される時と似たようなトランザクション・フローであり、システム・コールexitによって実現される。

5.3 シグナル処理及びアカウンティング処理

プロセス間通信の一種、シグナル処理をCOSMOSでサポートする場合、指定されたpidまたはpgrp(グループ名)に基づいて目的プロセスの所在を明らかにし、該当する場所へシグナルを送る必要がある。COSMOSでは、他プロセスにシグナルを送る要求はすべてネットワーク・マネージャに出す。ネットワーク・マネージャが対象プロセスを探し出してシグナルを伝える。そうすることによって、

- ・ネットワーク・マネージャによるネットワークワイドなプロセスのサーチで、同じグループに属するすべてのプロセスを探し出すことができる。

- ・pidが指定された場合プロセッサでそのプロセスのoidから存在場所を判別できるように思われるが、execを重ねることによって実際に居る場所はプロセスの元のpidで表されている場所と異なる場合も十分ありうる。ネットワーク・マネージャを通せば該当プロセスの居る場所が判る。

COSMOSでは、アカウント情報はすべてネットワーク・マネージャを通してexit時にファイル・サーバ上にあるアカウンティング・ファイルに書くことにする。そうすることによって、アカウンティング・ファイルへのアクセスの競合が避けられ、ネットワークワイドなアカウンティング処理ができる。

6 おわりに

以上でCOSMOSに必要なネットワーク管理機能及びそれらの実現方法について述べた。試作システムCOSMOS 1では集中管理方式を採用し、特有なノード構成、すなわちネットワーク内に唯一に存在するネットワーク・マネージャというノードを設けている。ほとんどのネットワーク管理機能はネットワーク・マネージャによって比較的小さいオーバーヘッドで実現されている。一方、より大規模なシステムの構築を考える場合、それらを更に機能別に分割することも可能である。

<参考文献>

1. 橋爪: 「ネットワーク資源共有のための統合型ネットワーク・オペレーティング・システムの研究」東京大学工学部 学位論文 1984年1月。
2. 計: 「分散型オペレーティング・システムにおけるネットワーク管理の一方式」東京大学工学部 修士論文 1986年3月。