

## 分散名前管理構成法に基づく論理シミュレーション

1D-3

吉宇田 フミ子 田中 英彦 元岡 達  
(東京大学 工学部)

## 1. はじめに

情報化社会を迎え、計算機網は重要性を増している。とりわけ、網での名前管理は基本的で重要なものである。このようなことから、これまでに、(1)名前を識別名と記述名(要素は意味付け名)の2面から捉えることを基本とし、(2)名前管理の基本として Pnd, Cnd, Undなどのネーミングドメイン(以下 n. d. と記す。)を導入し、(3)名前にケーバビリティを付加し、(4)相対名前付け方式を用いて、分散系の名前管理の一案を示してきた〔情全31, 60-5〕。本文ではこの構成案のシミュレーション結果を主に(4)に関して述べる。

## 2. 構成方式の概略

本文で置く仮定は、(1)名前とアドレスは〔Shoch78〕の定義に従い区別する。名前を用いた管理を調べアドレス管理は扱わない。(2)OSや通信制御など、すべてを複数のプロセスの集りとして捉える。(3)プロセスは、核、システム、ユーザの3種類とする。核プロセスとしての名前管理プロセスや通信管理プロセスはホスト毎に1組ずつ置く。(4)プロセスの名前付けについて調べる。

名前の概念的捉え方における利用者、識別名、記述名間の写像関係は図2-1となるが、構成案では分散系や管理効率を考慮して以下の(1)-(3)の制限を入れた。これに基づき、各種の表の配置を行い、名前管理をすることとした。

(1)写像 f を局所性管理の Pnd から PT (プロセステーブル) に限定する。(2)写像 g は Pnd と通信での名前を表わす Cnd を基に他の識別名 n. d. 間の写像関係を作る。(3)写像 u と v とは記述名 n. d. である ldes

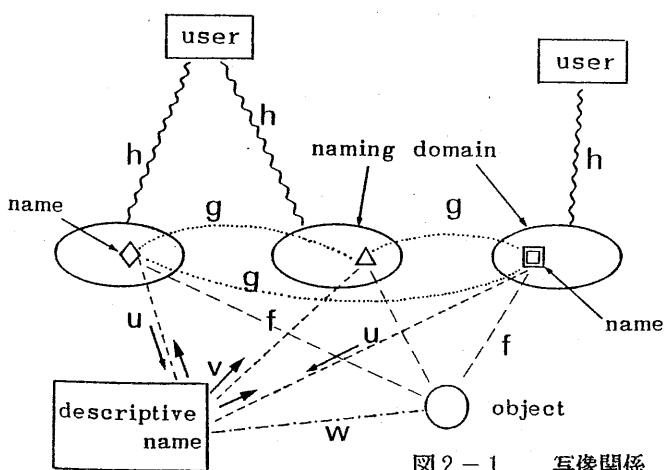


図2-1 写像関係

や rdes を基として g と同様な関係を付ける。

## 3. シミュレーション プログラム

名前管理処理手順を言語 Pascal で記述した。その構成は、動的名前付け処理と通信のための静的名前変換処理の2つに大別される。動的名前処理は新たに生じた対象に対する名前の登録・結合と、消滅した対象に対する削除手順であり、静的名前変換処理は結合された状態でユーザ間の通信時の名前変換をおこなう。通信はメッセージパケットを用いた転送として表現した。n. d. などの表は配列とレコードを組合せ名前管理や通信管理プロセスは手続き (procedure) とし、全体のスケジューラはメインプログラムで構成した。

動的名前処理手順は → や ← を通信、NAPi をホストiにおける名前管理処理プロセスとすると、  
利用者 → NAP1

NAP1 での処理 (フェーズ I)

NAP1 → NAP2

NAP2 での処理 (フェーズ II)

NAP2 → 利用者 2

NAP2 ← 利用者 2

NAP2 での処理 (フェーズ III)

NAP1 ← NAP2

NAP1 での処理 (フェーズ IV)

利用者 ← NAP1

となる。

## 4. シミュレーション結果

前章で述べたプログラムを東京大学大型計算機センタの VAX 11/780 を用いて実行時間、ステートメント数、n. d. などの表引きの回数を調べた。

プログラムの大きさは、ソースは注釈付きで 6600 行、オブジェクトは、プログラム領域 24 KB、データ領域が、295 KB となった。(UNIX の p i, p x コマンドによる。)

実行時間は time コマンドで測った。実行ステートメント数 (x) と時間 (y 秒) の関係は図4-1 のグラフより、ほぼ比例し

$$y = 1.5x (1/1000) + 1.7$$

である。

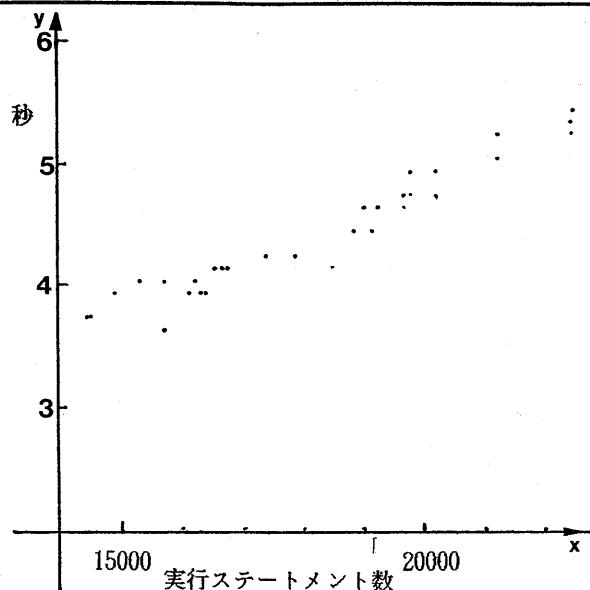


図4-1

### 実行ステートメント数 (x) と時間 (y 秒) の関係

処理プログラムは殆どが表サーチ、表の値の読み出し、書き込みであった。その結果を表4-2に示す。

## 5. 名前管理方式の検討

シミュレーション結果を基に名前管理構成法の複雑さや利用者に対する負担について、考察する。

表4-2よりプロセス間通信での名前変換は、n. d. 管理用のnamendと、名前変換に必要なn. d. (und, cnd, pnd) およびその間の写像 (nm, culr) であり、アクセス回数も最小回数である。名前の結合や削除処理においても、書き換えの必要のあるn. d. を直接引いている。このように名前変換は最大3回で済み、相対名前付け方式を探りながら、変換手順が簡単になった。この理由としては、写像gの決め方を分散系で不可欠な通信用のCndと局所性管理のPndを軸にして作成したことや、n. d. に写像gの一部を

表4-2 表サーチ、読み出し、書き込み回数

(注 c = 比較, r = 読出, w = 書込, s = プログラムソース行数, e = 実行ステートメント数)

含めた〔情全30,7P-2〕為、と考えられる。

C n d や P n d では表が比較的大きくなり、アクセス頻度が多いと考えられるが、これらの処理負荷や競合解決に関しては、P a s c a l ではこれらの点を記述しにくいことや、今回の目的が操作手順にあったことなどから、扱っていない。

図4-1と表4-2より、 $x$ と $y$ とは比例すると看做せるのでステートメント数で考察する。表は配列の形であり、表引きはwhile文を用いているので線形サーチとなり、表の後半にある要素を引く場合ステートメント数が増しているが、実装時には効率化手法が必要である。

利用者には、(1)U n dを設けたことで、利用者向きの独自の名前付けが可能になったことや、静的名前管理の内部構造はブラックボックスとして扱えること、(2)動的名前処理では、N A Pに依頼し、結果を待つだけでよいので負担は軽い、と言える。

## 6. おわりに

分散系向きの名前管理の一案を相対名前付けおよび写像について、論理シミュレーションをおこない、比較的簡単に構成できることを示した。

今後、ネームサーバや記述名との関係をより明確にするとともに、OSIのモデル上にこれらの機能を位置付けること等が課題である。

参考文献

## 〔情全31,60-5〕：吉宇田他、分散名前管理システムの構成法の比較と検討

情報処理学会第31回全国大会pp84-846

(情全30,7P-2) : 古宇田他, 分散処理システムにおける  
名前管理の一手法

情報処理学会第30回全国大会, PP1089-1090  
(Shoch78) : John F. Shoch : Inter-network Naming,  
Addressing and Routing, IEEE1978