

## Apostle の名前管理機構

田中啓介, 井田昌之  
青山学院大学  
情報科学研究センター 研究教育開発室

青山学院大学情報科学研究センターでは現在 Trinity 構想と呼ぶプロジェクトを推進している。Trinity 構想の中心は *Apostle* と呼ぶ分散処理機構である。*Apostle* ではその構成要素となっているホストが遠隔地に設置されている等の理由により疎結合状態にある場合にも、ローカル/リモートの違いを意識することのない統一された利用環境を提供する。IS (Information Server) は、この *Apostle* システムの上でユーザに関する名前情報を管理するためのデータベース機構である。本論文では、この *Apostle* の名前管理機構 IS について、その設計と実現方式を述べる。

IS は YP (Yellow Pages) に対して上位互換性を持たせた。*Apostle* は 64kbps の同期回線による結合をベースとして、三台の計算機を接続し、それらの中で同一の利用環境を提供する。このために IS は 1) ユーザ名をキーとする名前データのホスト間での共有、2) データ参照時のホスト間通信の抑制、3) 更新時の通信量の極小化、4) 特権ユーザの管理権限を分散したグループ管理、を実現した。

## The Information Server: A Name Management System for Apostle

Keisuke TANAKA, Masayuki IDA  
Computer Science Research Lab., Information Science Research Center,  
Aoyama Gakuin University,  
4-4-25 Shibuya, Shibuya-ku,  
Tokyo, 150, Japan.

The Trinity Initiative of Aoyama Gakuin University is a project of Information Science Research Center. A distributed system called '*Apostle*' is the kernel of the Trinity Initiative. *Apostle* provides an integrated computing environment to users among loosely coupled computers. IS (Information Server) is a subsystem of *Apostle*. IS is a database management system to manage name information for users. This paper describes a design and an implementation of IS.

IS is upward compatible with YP (Yellow Pages). *Apostle* is constructed over the connection of three computers in campuses of Aoyama Gakuin University. Each computer is connected with each other via 64kbps synchronous line. IS has the following features. 1) name information is shared with components, the key of data is 'user name', 2) transfer of name information over the connection of computers is not necessary on a name reference stage, 3) amount of transfer data is minimized in a data modify stage, 4) the right to manage the database is distributed to several managers of groups, each manager has the right to create/modify information of users belonging to his group.

# 1 はじめに

## 1.1 Apostle System

青山学院大学情報科学研究センター研究教育開発室では、Trinity 構想に基づくネットワークシステムの構築を進めている。この Trinity 構想は、青山学院大学の三キャンパスにまたがったネットワークの上で、統一された計算機利用環境を提供することを目的としている。その達成のために、以下の二つのシステムの構築を進めている。

1. Apostle と呼ぶ仮想システムの構築 [1,2]
2. Apostle を中心とするネットワーク環境の構築

Apostle は、青山、世田谷、厚木の各キャンパスに置かれたゲートウェイコンピュータ (Apostle では component computer と呼ぶ) 同士を接続し、これらを仮想的に一台のシステムに見立てる機構を提供する。そのために、データベース管理部 (IS)、ユーザインタフェース部 (UI)、File Cacher 部 (FC) を主な構成要素として設計を行なった。IS は統一環境を提供するためのユーザ情報を管理・提供し、UI はあたかも一台の計算機を使っているかのような利用環境を提供し、そして FC は利用するキャンパスと異なるキャンパスの component computer 上に存在するファイルをアクセスする機能を提供する。また、基本オペレーティングシステムとしては UNIX を採用した [1]。

本論文では、この Apostle の名前管理機構について述べる。

## 1.2 ネットワークにおけるユーザ管理

一般に計算機システムにおいて利用者の利用環境を提供するためには何らかのデータベースを必要とする。このデータベースによって利用者はファイルアクセスの権限を保証され、あるいは制限される。また、このデータベースの内容に従ってシステム内での利用環境が提供される。すなわち、オペレーティングシステムでのユーザ管理とはこのデータベースの管理である。

Apostle は component computer の間で統合された利用環境を提供する。そのために component computer 間で利用者情報を共有し、統一的に管理する機構を用意する。このためのデータベース管理機構が IS である。

## 1.3 実験環境

Trinity 構想の実験を行なう環境を Figure 1. に示す。図中各 component computer 間を結合している回線は

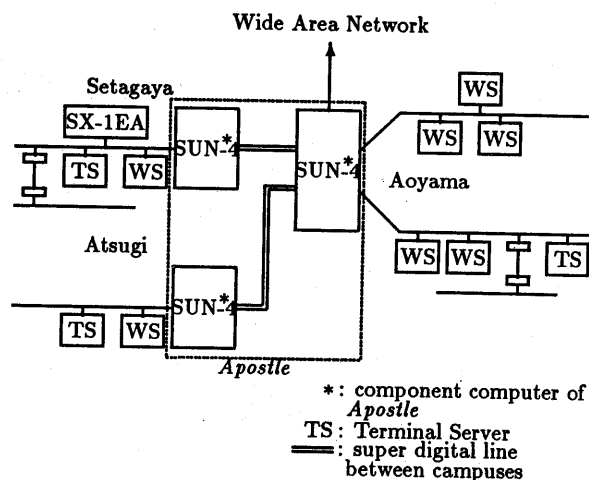


Figure 1: 実験環境

64kbps の同期回線であり、青山学院大学ではこれを高速デジタル回線によって実現している。

各キャンパスが位置的に離れていること、あるいはその間の回線にデータ転送量において制限があることから、IS などの Apostle 基本機能は component computer 間通信を最小限にとどめるように設計する。

## 2 Apostle の名前管理機構の概念

### 2.1 利用者情報のデータベースの一元化

IS は、component computer となっているホストの間でログイン名を統合管理し、データの参照を行なうホストに依存せず、[ログイン名] というキーに対して同一のデータを提供する。ここで参照されるデータはログイン名をはじめとする名前データであり、IS はログイン名を中心とした名前管理機構となる。

計算機を使用するユーザはログイン名 (ユーザ名) とパスワードを与えることによりそのシステムの利用を開始する。以後利用者のシステム上での行動は全てログイン名をキーとして与えられるデータによって決定される。すなわち、ネットワーク全体から見れば利用者情報のデータベースは [ログイン名, ホスト名] をキーとして持つ。しかし、IS では支配下のホスト間でログイン名を統一することで利用ホストとは無関係のデータベースを提供する。

## 2.2 管理権の分散機構の設定

IS では管理データの管理権を特権ユーザに集中させず分散するための機構を提供する。分散の単位としては UNIX の設定するグループを用いる。

UNIX においては、全てのデータ管理責任は特権ユーザに与えられている。これは各種データの集中的な管理が可能であった反面、ネットワークの発展に伴い以下の弊害が生じてきた。

- 作業の増大  
ネットワークが広範なものになればなるほど管理者としての作業が増加する。これは、逆に個々の管理作業に迅速な対応が困難になることとなる。
- 管理権の集中  
管理権の集中により管理上の不手際がネットワーク全体に、極めて危険な状態で波及することになる。

このような弊害を取り除くために、危険を分散し、個々の管理作業の負担を軽減する機構を提供する。このため、個々のデータの管理責任範囲として UNIX におけるグループの概念をそのまま利用する。

ここでは以下のような管理権の分散をはかる。

- 全体を統括する管理者を設定する。  
この管理者は *Apostle* システム全体が矛盾なく動作するための管理を行なう。具体的には、個々の管理責任者の設定やユーザに対するグループの設定を行なう。  
権限の集中を避ける意味で、この全体管理者と個々の component computer の特権ユーザとは別に設定する。
- グループ管理者を設定する。  
全体管理者はグループとその管理者を設定する。そのグループに所属するユーザに関する情報の登録/変更は、このグループ管理者が担当する。  
同一グループ所属のユーザは全て同じ GID を持つ。

システムの円滑な動作を保証するためグループ管理者の管理作業は全て自動的に全体管理者に報告される。また、グループ管理者はその責任範囲を越えて設定を行なうことは出来ない。例えば、グループ管理者はそのグループに許可された数以上のユーザを登録することは出来ない。あるいは、そのグループにファイルの作成が許可されていないようなディレクトリをホームディレクトリとして設定することは出来ない。このように報告の機構と管理上の制限を

設けることで、グループ管理者の行き過ぎた行為を防止する。

## 3 名前データベースの設計

### 3.1 データベースの構造

データベースにはユーザに関する情報が登録されるが、ユーザを束ねる存在としてのグループの情報も登録される。このデータベースは以下のような内容を持つ。

- グループに関する情報
  - グループ名, GID  
グループ管理の際の管理の単位であると同時に、UNIX の本来の機能によりファイル共有の単位としても扱われる。
  - 登録可能メンバ数  
グループ管理者が登録できるメンバの数を制限する。
  - グループ管理者の情報
  - 現在のメンバリスト
  - その他のグループ情報  
グループを表す部署・組織名などの補足情報
- ユーザに関する情報
  - ユーザ名, UID  
ログイン時にユーザが使用する名前である。他のユーザ情報を引き出すためのキーとなる。また、個々の component computer のオペレーティングシステムにおいてファイルアクセス時等の下位レベルでのユーザ識別のためには UID が用いられることからこれも統一管理する。
  - 参加グループ, GID
  - 課金情報  
IS が管理するのは課金情報のデータ自体ではなく、それを引き出すためのキーとなる情報(名前)である。
  - 個人情報  
ユーザの本名, 所属組織名, 住所等
  - パスワード情報
  - ホームディレクトリパス名
  - ログインシェル

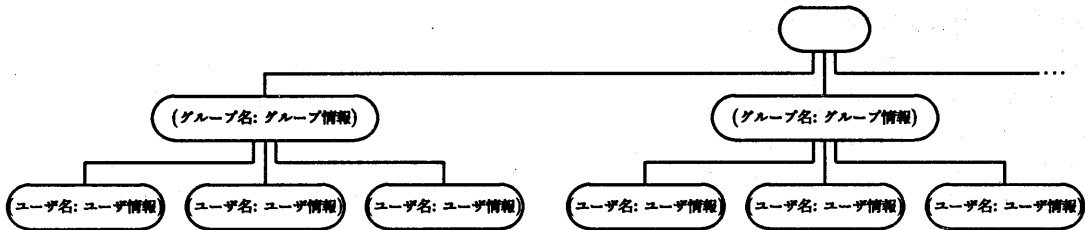


Figure 2: グループ情報とユーザ情報

— メール交換のための情報

メール交換の際の宛先として用いられるメールボックスの名前や、そのメールボックスが実際に位置するホストの名前など。

グループ情報とユーザ情報の間には Figure 2. に示す関係が存在する。グループにユーザ管理の役割を持たせたため、同一のユーザが複数のグループに属することは出来ない。

### 3.2 データベースの複製の設置

Apostle において各 component computer 間の結合は Ethernet ではなく 64kbps の同期回線である。また、この回線は Apostle を中核とするネットワーク全体のバックボーンとしての役割も持つ。そこで、IS 機能のための通信は極力減少させる。

また、IS のような利用者情報データベースはその更新に比べ参照の回数が圧倒的に多いと予想される。そこで、全く同一のデータベースすなわち一つのデータベースの複製を全ての component computer が保有するという形態を採る。これによりデータベースに対するオペレーションのほとんどを占める「参照」はローカルな component computer の内部でのみ行なわれる。

### 3.3 データ転送量を極小化するデータ更新

データ更新の機構は、各 component computer の複製に対して同時に行なう。この機構は、データの不整合を避けるために lock 機能を持つ。

一般に複製を持つ分散データベースの場合、データの更新方法は以下の二通りの方式がある。

- 一箇所のデータベースをマスタとし、このデータベースに対する更新を行なう。更新後のデータベースを他の component computer に配布する。

この場合、更新機構そのものは簡単なものになるが、データの不整合が生じ易く、あるいは配布の際の転送データ量が増大する。

- 同時に全ての component computer 上の複製に対してデータベース更新を行なう。

この場合、更新手続きが多少複雑になるが、データの不整合は生じない。また、更新の際のデータ転送量は最小限のものになる。

IS ではデータ転送量の極小化を第一に考え、後者の方式を採用した。

更新の機構はサーバ=クライアントモデルを利用して構築する。すなわち、各 component computer にはデータベースの複製を管理するサーバが存在する。更新を要求するクライアントは、これらに変更要求を行なう。

データ更新の権限のチェックは、クライアント、サーバで共に行なわれ、変更権限のない管理者、ユーザからの要求、あるいは変更できない項目に対しての変更要求は拒否される。

Apostle システムの各 component computer 上にデータベースの複製が置かれることから、データの更新を行なうクライアントは最初に全体に対して lock をかける。これにより、矛盾する変更が同時に生じることを防ぐ。また、各 component computer には優先度を設定し、同時に変更要求が発生した場合であっても dead-lock に陥ることを防ぐ。

さらに、万一の場合に備え、各 component computer の整合性をチェックする機構を備える。このチェックは定期的に各 component computer の持つ複製同士を照合し、矛盾がある場合、全体管理者に通知することで達成される。

## 4 YP を中核とした IS の実現

### 4.1 IS の管理するデータベースファイル

IS は以下の六種類の管理データベースを持つ。

- メールボックス情報 (aliases)
- グループ情報 (group)
- ユーザ情報 (passwd)
- 個人情報 (private)
- 課金情報 (account)
- グループ管理のための情報 (apostle\_group)

以上の管理データベースは YP (Yellow Pages) [3] の map として実現する。その実体は UNIX の標準ライブラリ DBM によって作成される DBM ファイルである。

aliases, group, passwd の三種は SUN の提供する YP に存在するが、他の三種類は存在しないので新設する。また、既存の三種は現在の YP の形式をそのまま使用する。したがって、IS は YP に対して上位互換性を持つ。

### 4.2 ypserv を使用したデータ参照

データベース参照のための機構としては YP におけるデータ参照機構である ypserv をそのまま利用する。したがって、Apostle の全ての component computer 上では ypserv が動作し、そこでは同一の DBM ファイルが管理される。

IS におけるデータ参照は以下の手順で行なう。

1. データ参照を希望するプロセス (client) は同一 component computer 内の ypbind を呼び出し、ypserv の動作するホストを探す。
2. ypbind は無条件で同一 component computer 内の ypserv の情報をデータ参照プロセスに与える。
3. データ参照を行なうプロセスは ypbind から得た情報を利用して ypserv に対して通信を行ないデータ参照を達成する。

以上のデータ参照手続きを Figure 3. に示す。

既存の YP では ypbind は broadcast 機能を用いることで、データベース参照に対して応答することの可能な ypserv を捜し出す。IS においては全ての component computer に ypserv が存在することから、このような ypbind の働きは不要であり、ypbind が存在する必要はな

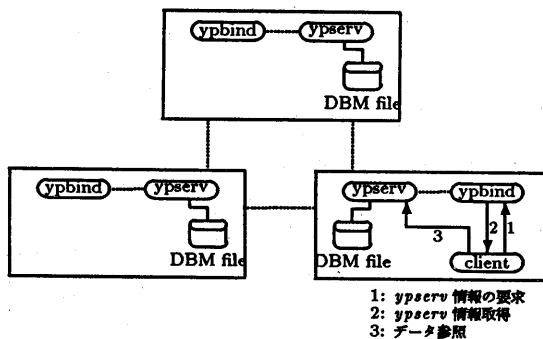


Figure 3: データ参照手続き

い。しかし、ypbind をなくすことで、既存の YP への参照を行なう多数のアプリケーションを変更することになるので、ypbind の機能を削減することで参照機能を実現した。

### 4.3 ypupdated によるデータ管理、データの更新

データの更新は、データ参照機構とは独立した機構として新設する。

データ更新は各 component computer 上の DBM ファイル管理サーバ ypupdated が担当する。データ更新は以下の手順で行なう。

1. データ更新を担当する client が起動される。  
この時点で、client を起動した利用者のチェックが行なわれ管理者でない、あるいは管理権限の及ばない変更は拒否される。
2. データ更新 client は、ローカルな component computer 上に存在する ypupdated と通信を行ない、全ての component computer の ypupdated の情報、及びその中で最も高い優先度を持つ ypupdated の情報を得る。
3. client は最高優先度を持つ ypupdated との通信を行ないデータの変更権を得る。  
この時点で他の client によるデータ変更が許可されている場合は変更作業は失敗する。また、変更要請のあったユーザ、変更内容のチェックが行なわれ権限のない変更は拒否される。データ変更権を得た場合、client はデータ変更用 ID を得る。
4. データ変更権を得た client は他の ypupdated に対

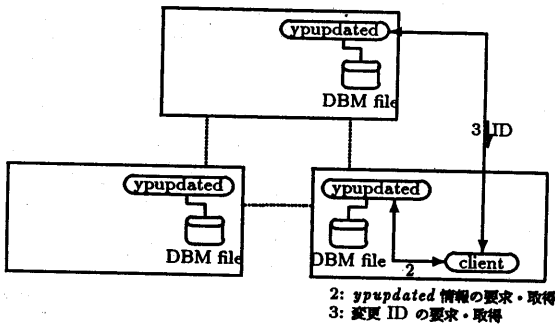


Figure 4: データ更新手続き (1)

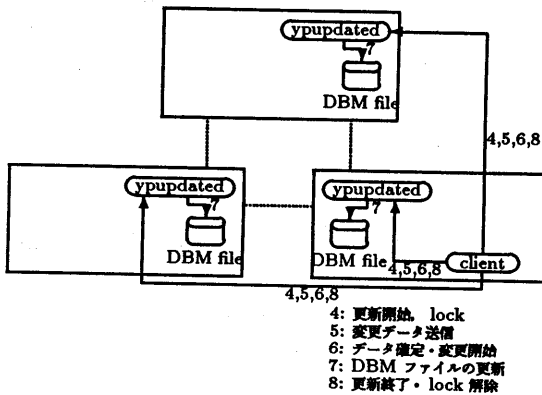


Figure 5: データ更新手続き (2)

してデータ変更用 ID を与えデータ変更を開始することを伝える。

5. 変更データを各 ypupdated に送る。
6. 変更データの送信が成功したことを確認後、client は各 ypupdated に対してデータ変更の実行を要請する。
7. 各 component computer の DBM ファイルが変更される。
8. client はデータ変更の lock を解除し、データ変更を終了する。

以上の手続きを Figure 4,5. に示す。

データ更新のための client としては、Table 1. に示すものを準備する。

#### 4.4 ユーザ管理のための補助機能

新規のユーザ登録ではホームディレクトリの作成が必要となる。しかし、あるユーザに対するホームディレクト

Table 1: データ更新用の client

newgroup:	グループ登録のための client
modgroup:	グループ登録内容変更のための client
delgroup:	グループ登録抹消のための client
newuser:	ユーザ登録のための client
moduser:	ユーザ登録内容変更のための client
deluser:	ユーザ登録抹消のための client

リを作成し、その所有者を変更する作業は各システムの特権ユーザにしか許されていない。

そこで、ホームディレクトリ管理のためのサーバ rsetupd を特別に設けホームディレクトリの設置、所有者の変更を行なう。

rsetupd は新規のユーザ登録が発生した時点でデータ更新用 client から呼び出され、以下の処理を行なう。

1. 処理要求の正当性を確認する。
2. client はホームディレクトリのパス名、所有者の UID, GID を rsetupd に送信する。
3. ホームディレクトリに相当するディレクトリを作成する。
4. .cshrc や .login などのスタートアップファイルを作成する。
5. 作成したファイル・ディレクトリの所有者 UID, GID を指定のあったものに変更する。

#### 4.5 スーパーコンピュータの管理との整合性

Trinity 構想によるネットワークシステムが構築された時点で、ネットワーク上に存在するスーパーコンピュータと Apostle は協調した管理を行なう予定である。したがって、IS の管理データにおける課金情報としてスーパーコンピュータのアカウントコードと同形式のものを採用した。

## 5 まとめと評価

### 5.1 まとめ

今回設計した IS は、Apostle での利用を第一に設計されているが、データベースを管理する目的に対しては一般のネットワーク上でも十分に有効である。この IS で特徴的なことは以下の通りである。

- ネットワーク上で統一的なデータベース管理を可能とする

- データ参照の際にはネットワークを使用した通信は生じない  
各 component computer はある一つのデータベースの複製を保持している。各 component computer で発生するデータ参照はその component computer 内で処理する。これは、component computer 間の通信媒体に十分な帯域が保証されていない *Apostle* では重要である。

- 矛盾のない更新

データベースは分散管理されるが、同期をとって更新されることから、複製の間に矛盾は生じない。但し、データ更新時にデータベースの複製を保持する全てのホストが動作していなければならないという条件が存在する。

- 各ホストの管理形態には基本的に依存しない

データベース上に存在する情報は基本となる名前情報のみであり、その情報をいかに利用するかは各ホストの自由である。従って、本質的に本データベース管理機構は動作オペレーティングシステムには依存しない。しかし、現状ではこの IS は *Apostle* のために用いられるので全ての component computer でその解釈、利用方法は同じである。

また、その実現の上での特徴として、UNIX のグループをデータ管理・更新の権限の範囲を規定するものとする定義を与えたことがある。これによって、ユーザのグループ管理が可能となった。そのため、特権ユーザへ集中していた権限は分散され、権限集中によるシステムへの悪影響は排除された。これは、利用ユーザ数が多数となるような UNIX システムでは重要な機能である。

また、実験段階にあつて、以下の問題点が明らかになったので、その対策を進めている。

- セキュリティ

IS の基本機能で、データ変更の権限はチェックしているが、client 側からの虚偽の申請に関してはこれを防ぐ手段を持たない。このため、悪意のユーザが作成した client に対して、本来の権限以上のデータの変更を許可してしまう可能性がある。

この問題に対しては、サーバからのパスワード要求や client からのコールバックを利用した権限の厳密なチェックを行なうことで解決をはかる。

- ネットワーク全体の協調

Trinity 構想では *Apostle* を中心とするネットワー

ク上の全てのシステムの間で統合的な管理を行なうことをその目的の一つとしている。そのために、IS よりももっと広い範囲、すなわちネットワーク全体でのデータベース管理機構の設計を進めている。このデータベース管理機構は *Apostle* のように完全なシステムの統一を目的とせず、各システムの自主管理の部分を残すものとなっている。IS とこのデータベース管理機構との協調が必要になるため、今後、データ更新部を中心として変更を加える予定である。

## 5.2 現状及び今後

実験は二段階に分けて行なわれている。

1. 二台の UNIX マシン間による基本機能の実験

Ethernet で結合された二台のマシン間で IS の基本機能の実験を行なう段階。

2. 三キャンパスに展開した実験環境での実験運用

現段階では、ユーザ数・更新頻度が限られており十分な検討データは得られていない。今後の実験運用によってこれらが明らかになると考えられる。

今後の課題としては、実験運用の結果から公開運用に向けてのシステムの修正を行なうことが挙げられる。ここでは、セキュリティの問題の解決が重要な点の一つとなる。また、実験環境と類似した状況、すなわち遠隔地との疎結合にある LAN 同士で情報の共有をはかるための機構としての一般化も進めていく。

## 謝辞

*Apostle* の実験に対し多大な御尽力をいただいている情報科学研究センター所長 大矢知浩司教授、並びにセンター諸氏に感謝致します。

## 参考文献

- [1] Ida, M. and Tanaka, K., 'The Harmonic Connection Concept in the Trinity Initiative of Aoyama Gakuin University', Proc. of JCCW'88, pp111-120, Jul. 1988.
- [2] Ida, M. and Tanaka, K., 'The Apostle System Overview', CSRL Technical Report #88-001, Aug. 1988.
- [3] Sun Microsystems Inc., 'The Yellow Pages Protocol Specification', Feb. 1986.