

3T-9

LISPマシンELISにおけるNFSの実現

中山千代美* 大江哲男** 長坂篤**

* (株)沖テクノシステムズラボラトリ ** 沖電気工業株式会社

1 はじめに

近年のワークステーションとローカルエリアネットワークに基づく開発環境の普及には目覚ましいものがある。特にこの環境におけるUNIXシステムの役割は大きく、非UNIXシステムにおいてもUNIXシステムで用いられているネットワークプロトコルやネットワークアプリケーションを備えることは、ネットワーク資源の有効利用を計る上で非常に重要である。ここで取り上げるNFS(Network File System)^[1]はUNIXシステムを中心にいくつかのシステムで稼働している分散ファイルシステムであり、ネットワーク環境においてファイルシステムの共有を可能にする。

本稿では、LISPマシンELIS上へのNFSのインプリメントを通して、非UNIXシステムにNFSを実現する場合の問題点を指摘し、その解決法を提案する。

2 NFSシステムの概要と構成

NFSはマウントの概念をネットワーク上に拡張した分散ファイルシステムである。関連するプロトコル、サーバ、ファイルシステムの関係を図1に示す。トランスポート/ネットワーク層としてUDP(User Datagram Protocol)/IP(Internet Protocol)が使われる。RPC(Remote Procedure Call)^[2]は、通常の手続き呼び出しとよく似ており、引数と結果をネットワーク上でやり取りする事でプロセス間の同期を取る通信手段である。このときネットワーク上を移動する引数と結果はXDR(eXternal Data Representation)^[3]に従ったデータ表現をとる。

NFSの動作シーケンスは以下の通りである。まず、マウントアクセス関数を通じてマウントサーバにアクセスすることによりリモートマウントを行い、マウントディレクトリのファイルハンドルを獲得する。このときマウントサーバの待つポート番号はポートマップによって事前に知ることができる。以後、このファイルハンドルを基に、NFSアクセス関数からNFSサーバを用いてリモートファイルシステムを操作する。そしてシステム終了時にマウントアクセス関数を用いてアンマウントする。

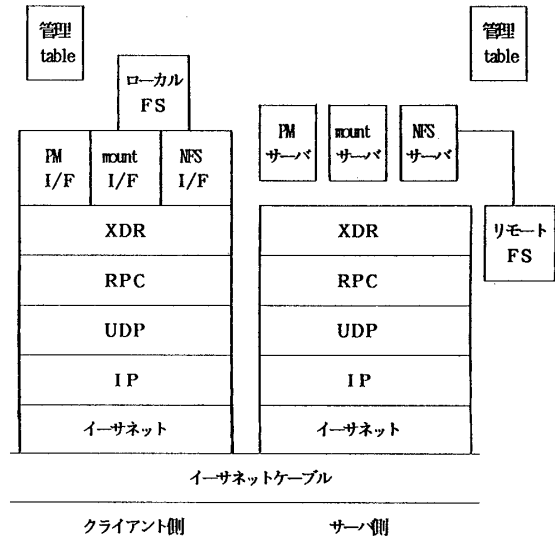


図1. NFSシステムの構成

3 実現上の問題点

3.1 ファイルシステム

NFSはファイルの表記法において特定のOSに依存しないが、他の機能はUNIXシステムを想定している。ELISファイルシステムは、以下の点においてNFSと異なる。

- (1) ELISファイル管理機能：ELISファイルシステムは、ファイルのソフト削除、ソフト削除されたファイルの復活、ソフト削除されたファイルの抹消、バージョン管理等の機能を持つ。
- (2) シンボリックリンク、ハードリンク：ELISファイルシステムはこれらの機能を持たない。
- (3) ファイル属性：ELISファイルシステムは、バージョン保存数、自動ファイル抹消時間間隔、削除不可、書き換え不可等の属性を持つ。しかし、ユーザID番号、グループID番号、ファイルシステムID番号、ファイルID番号等の属性は持たない。
- (4) ファイルネーム長：ELISファイルシステムのファイル名は任意長である。
- (5) 特殊文字：ELISファイルシステムのファイル名では、ファイル、タイプ、バージョンの区切り記号としてピリオド(.)を使用する。

3.2 認証

現在のNFSは認証としてUNIX認証を用いている。UNIX認証では要求メッセージにクライアントプロセスのユーザID番号、グループID番号を含む。しかし、ELISのプロセスは、ユーザID番号、グループID番号を持たない。

Implementation of NFS on Lisp machine ELIS
Chiyomi Nakayama*, Tetsuo Ooe**, Atsushi Nagasaka**

* Oki Technosystems Laboratory, Inc.

** Oki Electric Industry Co., Ltd.

3.3 既存UNIXシステム

既存UNIXマシンのNFSシステムには、"."ファイルと".."ファイルの存在を仮定した動作をとるものがある。

4 実現方法

ELISはファイルシステム、ネットワークシステム^[4]ともにLISPによって記述されている。従ってELIS/NFSもLISPにより以下の機能を追加変更することで実現した。

4.1 XDR/RPC

XDR変換関数は、NFSに必要な整数型、論理型など基本的データタイプのシリアライズ/デシリアライズ機能を提供する。より複雑な変換関数は、これら基本変換関数の組み合わせにより作成できる。RPCクライアント関数は、引数をシリアライズしそれをサーバへ送信する。そしてサーバからの応答メッセージをデシリアライズし結果として返す。RPCサーバ起動関数は、汎用のサーバプロセスを生成する。このプロセスは個々のサービスのディスパッチ関数を与えられることで目的のサーバとなる。

これらの関数を使用することで、RPCプロトコルの詳細を知らなくてもRPCアプリケーションを作成できる。

4.2 ポートマップ/マウント

ポートマップサーバ、マウントサーバへのアクセス関数群を提供した。クライアントはこれらの関数を通してポートマップ、マウント/アンマウントなどの処理を行う。それぞれのサーバは、リモートホストからの検索要求に対して応答する。

4.3 管理テーブル

(1) エクスポートデバイステーブル：エクスポートデバイスごとに、リモートマウントを許可するホストを記述する。

(2) ユーザテーブル、グループテーブル：ユーザ名とユーザID番号、グループ名とグループID番号の対応をそれぞれ記述する。このテーブルでマッピングをとることで、UNIX認証の問題とファイル属性の問題の一部を解決する。

4.4 NFS

NFSサーバへのアクセス関数群を提供した。クライアントはこれらの関数を通してリモートファイルシステムにアクセスする。NFSサーバはNFSクライアントからの要求を、ELISファイルシステムが解釈できる形式に変換し実行した後、結果をNFSクライアントの解釈できる形式で返す。

3で示した問題点に対するELIS/NFSの対処法を以下に示す。

(1) ELISファイル管理機能：NFSはこれらの機能をサポートしないので実現できない。現在のELIS/NFSサーバでは常に最新バージョンが処理の対象となる。また、ファイルの抹消はサーバ側で責任を負う。すべての機能を実現する

にはNFSプロトコルを拡張する必要がある。

(2) シンボリックリンク、ハードリンク：実現していない。ファイルシステムの内部構造の違いにより実現は難しい。

(3) ファイルID番号：ファイルがアクセスされるたびに動的にファイルID番号を生成し、これをメモリ上で管理する。

(4) ファイルネーム長：NFSのファイル名長255バイト、パス名長1024バイトという制限に従う。

(5) "." ".." ファイル：UNIXのセマンティクスに沿った処理を行う。しかしこれはNFSサーバで処理するより、UNIXシステムのNFSクライアントで対処すべき問題である。

4.5 ローカルファイルシステム

ELISローカルファイルシステムからリモートファイルシステムを透過的にアクセスできるように拡張した。拡張部分では、ELISファイルシステムとNFS間でデータの変換を行う。リモートファイルシステムのファイル名に含まれる"."は、ELISファイルシステムでは"\$"に変換して扱い、特殊文字(.)の問題を解決する。

また、パス名の中に明示的にホスト名を指定することで、リモートファイルシステムを自由にアクセスできる(オートマウント)機能も実現した。

5 今後の課題

- (1) ELISファイルシステムのNFSに対する最適化
- (2) ELIS/NFSの拡張
- (3) RPCアプリケーションジェネレータの整備
- (4) ブロードキャストRPCのサポート

6 おわりに

今回、LISPマシンELIS上にNFSクライアントとサーバを実現した。非UNIXシステムにNFSシステムを構築する場合の最大の課題は、ファイルシステムの違いをいかに効率よく吸収するかである。現在、ELIS/NFSは、SUN、NEWSなどのUNIXワークステーションとの間で動作している。速度は、ローカルファイルシステムには及ばないが十分実用に耐えうるものである。

参考文献

- [1] Sun Microsystems, Inc.: "Network File System Protocol Specification"
- [2] Sun Microsystems, Inc.: "Remote Procedure Call Protocol Specification"
- [3] Sun Microsystems, Inc.: "External Data Representation Protocol Specification"
- [4] 村上: "LANプロトコルのオブジェクト指向による実現", 情報処理学会第34年全国大会 1P-7, 1987