

NFS 高速化手法

3M-1

小澤みどり 琴屋秀平
 (株) 東芝 府中工場

1. はじめに

ネットワークに接続された計算機間でファイルを共有するNFS(ネットワークファイルシステム)において、NFS処理の一部を専用プロセッサで処理することによって高速化を図ったネットワーク制御プロセッサを開発したので報告する。

2. NFSの概要と性能

2.1 NFSの概要

NFSは1985年にSun Microsystems社によって開発された分散ファイルシステムである。クライアントとなる計算機からサーバのディスク上に存在するリモートのファイルも、ローカルなディスク上に存在するファイルも、その位置を意識することなく同一の手順でアクセスできることを特徴としている。NFSは分散処理を実現する中核としてメインフレームからPC(パーソナルコンピュータ)まで多くの計算機に実装されてきている。

NFSはRPC(リモートプロシジャコール)によって実現されている。RPCは通常プロセス内で行われているプロシジャ呼び出しを計算機間に拡張したものである。RPCのクライアントは実行すべきプロシジャ(プログラム番号、バージョン番号、プロシジャ番号で表現)と必要な引数をサーバに送り、応答を待つ。サーバでは指定されたプロシジャを実行し、その結果をクライアントに送り返す。この時、クライアント・サーバ間で送受信されるデータは固定小数点数のバイトオーダやMSB/LSBの位置や浮動小数点数の形式などが予め定められた形式に変換される。この機構はXDR(外部データ表現)と呼ばれている。

NFSでは下位層のプロトコルとしてUDP/IPを採用している。UDP(ユーザデータグラムプロトコル)はコネクションレス型トランスポートサービスであり、高速で実装し易い反面、データグラムの送達確認、再送制御、順序制御などは行われない。そのためNFSではRPCでこれらの制御を行い、確実なデータ転送を保証している。(図1)

NFSは米国におけるSun Microsystems社の登録商標です。

2.2 NFSの性能

NFSクライアントとなる計算機の高速化、ネットワークを含めたシステムの大規模に伴い、NFSサーバとして動作する計算機にはより一層の高速性と保安全性が求められている。

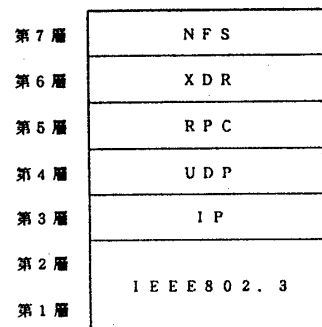
NFSによるリモートファイルアクセスでは以下のような点が、ローカルファイルアクセスとは異なり、オーバーヘッドとなる。NFS処理を高速化するためにはこれらのオーバーヘッドの削減/解消が必要となる。

(1) RPC/XDR処理によるオーバーヘッド

3つの番号で表現されるRPCと実際のプロシジャコールとの対応表を持ち、プロシジャへのデータをネットワーク上のデータ表現に合わせるためのXDR処理をクライアント/サーバの双方で実行する必要がある。特に計算機のバイトオーダがネットワーク上の形式と異なっている場合、XDR処理のコストが大きくなる。

(2) 通信処理(UDP/IP)のオーバーヘッド

UDP/IPはコネクションレス型のプロトコルであり、それ自身のオーバーヘッドはTCP/IPのようなコネクション型のプロトコルと比較して大きくはない。しかしUDP/IPでは1回に転送できるデータサイズは最大8KBであり、さらにUDPデータは使用する物理媒体の制限により複数のIPデータグラムに分割されて送受信される。このような場合、複数のIPパケットからUDPパケットの再構成を行う必要がある。これらのプロトコル処理はデータの移動を伴わない場合でも大きなオーバーヘッドとなる。



NFS Acceleration Method
 Midori OZAWA, Shuhei KOTOYA
 TOSHIBA CORPORATION, Fuchu Works

図 1

(3) 通信媒体のデータ転送速度

現在、広く用いられている物理媒体はCSMA/CD方式のバス型LANであり、データ転送速度は10Mbit/sである。バスに複数の機器が接続された場合、物理媒体上の衝突などによりデータ転送の実効速度は約300kByte/sとなる。

3. NFSプロセッサ

NFSプロセッサはNFSのサーバ機能を高速に処理する目的で開発された。(図2)

NFSプロセッサでは、従来CPUで行われていたUDP/IP処理、RPC処理、XDR処理をファームウェア化している。

NFSを実現するRPCはそのプロトコル(UDP/IP)と使用するポート番号(2049)により他の要求と区別できる。NFSプロセッサは、受信したパケットのプロトコルおよび宛先ポート番号を確認して、スイッチングモジュールによりNFSのRPC要求パケットとそれ以外のパケットを分別する。

一つのRPC要求を運ぶUDPパケットが複数のIPパケットにフラグメント化されて送信されてきた場合、まず発信元アドレス(IPアドレスおよび物理アドレス)とIPパケット識別子をフラグメントパケットテーブルに記録しておく。これは、IPパケット識別子によってUDPパケットの再構成を行うためである。

IPパケットがフラグメント化されている場合、UDPヘッダは最初のIPパケット内にしか存在せず、それ以降のIPパケットにはヘッダ情報は含まれていない。

NFSプロセッサ内で、正確にUDPパケットの再構成が行われるように、パケットを分別する時にこのテーブルが参照される。全てのフラグメントされたIPパケットが到着したならば、UDPパケットの再構成が行われ、該当するIPパケットのデータはテーブルから削除される。

完成したUDPパケットはならばRPCの識別とXDR処理を行う。XDR処理が終了したならばRPC種別とデータをドライバ経由でソフトウェアへ送り、専用のデーモンプロセスでサーバ処理(ファイルアクセスなど)を行う。

サーバ処理の結果はデーモンプロセスからNFSプロセッサへ送られ、XDR処理およびRPC処理が行われRPC応答パケットが作成される。この応答パケットはUDP/IPにて発信元へ返される。

NFSのRPC要求パケット以外の受信パケットに対しては従来のLAN制御プロセッサと同様に、従来のインタフェースを通してパケットをCPUに送り処理をする。また、ブロードキャストパケットやARPに対する処理も従来の機構をそのまま利用する。

4. 効果

NFSプロセッサを採用したことにより以下の効果が得られた。

(1) NFSサーバ処理の高速化

NFS処理専用のプロセッサとして高速なRISCプロセッサを採用したことにより、1個のRPC処理に要する時間がCPUで処理した場合と比較して30%短縮され、RPCの応答性能が向上した。

(2) CPU負荷の軽減

従来の処理ではCPUで実行されていたUDP/IP処理、RPC処理、XDR処理をNFSプロセッサで処理できるため、CPUの負荷が約40%軽減され、サーバ計算機上で他のアプリケーションのために多くのCPU時間を割り当てることが可能となった。

(3) 大規模化に対応

複数のNFSプロセッサを用いることによって、CPUの負荷を上げることなく複数のLANをサポートすることが可能となった。

5. おわりに

NFSプロセッサを実装することによって高性能なサーバを実現でき、大規模なシステムに対応することが可能となった。

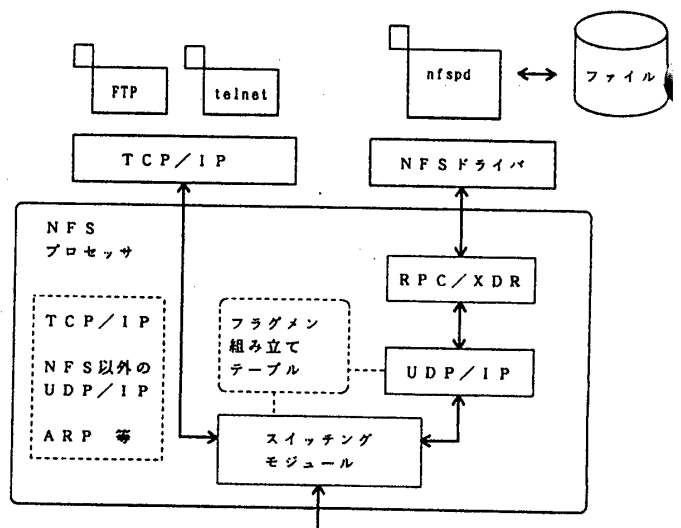


図 2