

分散サーバによるネットワークファイルシステムの研究

6 D-4

友平敏男

上原敬太郎

千葉滋

益田隆司

日立電子サービス(株)

東京大学 理学部 情報科学科

筑波大学 電子・情報工学系

東京大学 理学部 情報科学科

1 はじめに

単一サーバの分散ファイルシステムではファイルサーバに負荷が集中し、高速化の進んでいるネットワークを十分に活用できなくなる。本研究では、ファイルサーバを複数にし、サーバ間でファイルの複製を行なうことでクライアントからの要求の分散を図る。サーバ間のファイル一貫性維持プロトコルは弱いものにし、一貫性保持の処理を軽減する。本研究ではネットワークファイルシステム(NFS: Network File System)を拡張し、現行のものとの比較を行なった。以下、現行のNFSの基本的な動作とマルチサーバ化した場合の動作について説明し、実験の結果からファイルの読み込みに関してマルチサーバ化することの有効性を示す。

2 現行のNFS

現行のNFSでは特定のファイルに対するアクセスサービスを行なうファイルサーバは単一なので、そのサーバが持っているファイルへのアクセス要求が増えてくるとクライアントから見たファイルアクセスの性能の低下が大きくなる。以下に現行のNFSの基本的な動作を述べる。

● ファイルハンドルの取得 (lookup)

クライアントがファイルにアクセスする場合には、まずファイルをオープンしなければならない。この時にそのファイルのパスをlookup要求としてサーバへ送り、パスに対応するファイルハンドルをサーバから取得する。ファイルハンドルはサーバ上で作成され、サーバ上のファイルをユニークに識別する情報を持っている。以降、ファイルのアクセスはこのファイルハンドルを用いて行なわれる。

● ファイルの読み込み (read)

クライアントはファイルハンドルとアクセス開始位置を示すoffsetなどアクセスに必要な情報

をread要求としてサーバへ送り、サーバはその要求に対する処理を行なって結果をクライアントへ返す。

● ファイルへの書き込み (write)

readと同様の情報に加えてファイルへ書き込むデータなども含めてwrite要求としてサーバへ送り、サーバはその要求に対する処理を行なった後、クライアントへ処理終了を知らせる。

3 NFSサーバのマルチサーバ化

ファイルサーバへの負荷集中による性能低下の問題を解決するために、NFSサーバをマルチサーバ化する。マルチサーバ化するには、上の節で述べた基本動作をマルチサーバ対応の動作に変更しなければならない。その際に以下のような問題がある。

● ファイルハンドルの取得

ファイルハンドルがサーバ上で作成されるため、同じパスのファイルであってもサーバが異なるとファイルハンドルが違ってしまふという問題がある。解決方法としては以下の2つが考えられる。

1. クライアントが全てのサーバへlookup要求を行ない、対応するファイルハンドルを持つ。
2. クライアントは全サーバに対して共通なキーを持ち、サーバ側でそのキーから特定のファイルハンドルへ変換するような機構を設ける。

本システムではマルチサーバ化することの有効性をファイルの読み込み及び書き込み動作について確認するということに重点をおくため、比較的実装が容易な1の方法を採用している。しかし、この方法ではlookupのコストが現行のNFSに比べてサーバの台数に比例するコストがかかってしまうことになるので、適切な方法の検討が今後の課題である。

● ファイルの読み込み

クライアントがファイルを読み込むには、要求を出すサーバを選択する必要がある。現在のプロトタイプでは複数のサーバからランダムにひとつのサーバを選択している。しかし、本来は

A Study of Network File System with Multiple Servers.

Toshio Tomohira, Keitaro Uehara, Shigeru Chiba and Takashi Masuda.

Hitachi Electronics Services Co., Ltd., Dept of Information Science, University of Tokyo., Institute of Information Sciences and Electronics, University of Tsukuba., Dept of Information Science, University of Tokyo.

サーバの負荷やファイルのアクセス状況などに応じて適切なサーバを選択すべきである。適切なサーバを選択する方法は今後の検討課題である。

● ファイルへの書き込み

どれか1つのサーバで共有ファイルに対して書き込みがあった場合には、他の全てのサーバに対してその書き込みの内容を反映させる必要がある。このファイル一貫性保持の問題については、以下の解決方法がある。

1. 書き込み中のファイルに対する処理を行なうサーバを固定にする。別のサーバへ要求が来た場合には書き込み処理を行なっているサーバへ要求の転送を行なう。書き込みが完了したら全てのサーバへファイルを複製する。この方法では、クライアントから見た書き込みの処理性能は単一サーバ時と同等かそれ以下になると考えられる。
2. クライアントが全てのサーバへ要求を出す。クライアント側の負担が大きくなるが、サーバ側は単一サーバ時と同じ動作で済む。この方法では、クライアントから見た書き込みの処理性能は単一サーバ時よりも悪くなると考えられる。
3. 要求を受けとったサーバが他サーバへ要求を展開し、同時に書き込み処理も行なう。ファイル一貫性保持をサーバ側のみで行なうのでクライアントの負担はそれほど大きくはならないが、サーバ側の負担が大きくなる。クライアントから見た書き込みの処理性能は同時にアクセスしているクライアント数が増えれば単一サーバ時よりも若干良くなると考えられる。

クライアントから見た書き込みの処理性能が最も良いものは、3つめの方法である。このため、本システムでは3つめの方法を採用することにした。

4 実験

Sun SPARCstation/ELC上でNetBSD 1.2.1のNFSをベースとして実装を行なった。なお、現段階ではサーバは2台固定でlookupとreadの機能のみ実装が完了している。writeの機能については現在、実装中である。

このシステムと通常のNFSでreadの実験を行なった。実験環境は以下のようになっている。

サーバ(2台):Sun SPARCstation/ELC
CPU MB86903(33MHz,21Mips),mem 28MB
クライアント(1台):Sun SPARCstation/ELC
CPU MB86903(33MHz,21Mips),mem 40MB
ネットワーク:ethernet,10Mbps

実験の内容は、複数のプロセス(1個から40個)が同時にサーバのファイルアクセスするもので、それぞれのプロセスの処理時間を測定し同時に動作していたプロセス数で平均を取るというものである。各プロセスはファイルを1つオープンし、256バイト単位でreadシステムコールを2048回発行し、0.5MBのシーケンシャルアクセスを行なう。実験結果を図1に示す。

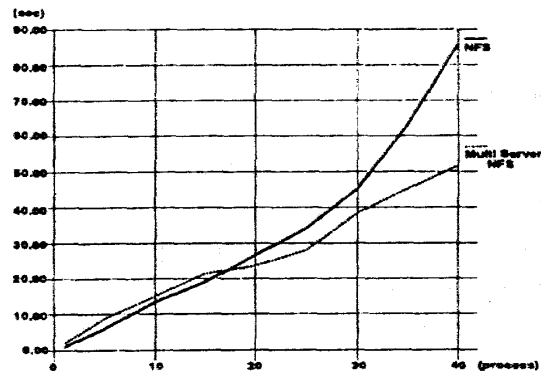


図1 read 処理性能

図の縦軸が処理時間で横軸がプロセス数である。実験の結果、NFSに比べて本システムの方がプロセス数の増大に対する処理時間の増加が少ない。この結果より分散ファイルシステムを単一サーバからマルチサーバ化することが読み込みにおいては有効であると言える。

5 おわりに

単一サーバの分散ファイルシステムではファイルサーバに負荷が集中し、極めて広い帯域を持つネットワークを十分に活用できない。ファイルサーバの負荷を軽減するために、ファイルサーバを複数にし、サーバ間でファイルの複製を行なう方法がある。本研究では、NFSをマルチサーバに拡張したシステムの実装について検討し、これまでに実装できたファイルの読み込みに関しては実験によってその有効性を示した。今後はファイルへの書き込みについても実装を行ない、その有効性を確認する。

参考文献

- [1] Hal Stern 著, 倉骨 彰 訳, "Managing NFS and NIS", O'Reilly & Associates, Inc., 株式会社アスキー, December 1992.
- [2] ELIEZER LEVY and ABRAHAM SILBERSCHATZ, "Distributed File Systems: Concepts and Examples", ACM Computing Surveys, Vol.22, No.4, December 1990.
- [3] Mahadev Satyanarayanan and James J. Kistler and Puneet Kumar and Maria E. Okasaki and Ellen H. Siegel and David C. Steere, "Coda: A Highly Available File System for a Distributed Workstation Environment", IEEE Transactions on Computers, April 1990.