ファイルサーバの仮想化方式に関する一検討*

6J－04

山川 聡　桂島 航　石川 潤　菊地 芳秀
NEC インターネットシステム研究所

1．はじめに

計算機の付属装置という位置づけにあったストレージは、近年の情報の多様化、大容量化を追い風として、SAN (Storage Area Network) やNAS (Network Attached Storage) といったネットワークストレージとして、システムの構築上、重要な役割を果たすようになり、急速に一般社会に普及し始めている。現在 NAS は、主に導入・管理が容易なファイルサーバとして、様々なシステムで運用されているが、その反面、各 NAS が独立したサーバとして稼動するため NAS 間での効率的なデータ運用などの連携が難しいという問題点があった。本稿では、複数の NAS を仮想化、統合して NAS 間の連携を実現し、これを一元管理するための一手法を提案する。

2．サーバ・スイッチングの現状と課題

ネットワークストレージ（NAS や SAN に接続されるディスクアレイ）が、今後、より広く普及していくためには

- ノード数が増えても管理が複雑にならない
- ノード数に比例して、容量や性能がスケーラブルに向上する(スケールアウト性)

ことが必須である。この要求に応えるべく、負荷分散、サービスの仮想化、ストレージの仮想化などのキーワードのもと様々なサーバ・スイッチング方式が提案されている。詳細は[1]に譲るが、例えば商用レベルでは、TCP などのトランスポート層ヘッダの内容をもとにクライアントからの要求を適切なサーバに振り分ける L4（レイヤ 4）スイッチにはじまり、IP ネットワーク上に名前空間（メタデータ）を管理する装置を配し、ここで適切なディスクアレイを選択させ、データはブロックアクセスで高速に取り扱うもの[2]など様々な手法が提案されている。また、文献[3]では、[2]を拡張し、データサイズによってブロックアクセスとファイルアクセスを切り替える方式などが提案されている。しかしながら、L4 スイッチには管理を容易にするような機能はなく、また、文献[2]や[3]の方式は out-of-band と呼ばれスケールアウトには有効だが、クライアントにも特殊なエージェントをインストールする必要があるなど、既存環境との融和性に課題が残されていた。

特に NAS に関しては、NAS 追加時のクライアントへの環境再設定の実行や、NAS ごとのリソース格納ポリシーの管理などの煩雑な作業を解決する手段はなく、NAS を統合仮想化することに対する要求が非常に高かった。これらの問題を解決する手段として、分散ファイルシステムを用いて密結合して動作する NAS 製品[4]も登場しているが、異種 NAS 間での相互接続ができないなど、やはり既存環境での相互運用に課題を残していた。

本稿では NFS（Network File System）バージョン 2、3[5][6]をファイルアクセスプロトコルとして用いた環境において、複数の NAS を仮想化して NAS の連携管理を可能とし、新規 NAS の追加やリソースの再配置などを NFS クライアントに対して透過に実行できる装置の一構成法について提案する。提案手法は、NFS クライアント-NAS 間のファイルアクセス要求と応答を非常に簡単な処理で中継する NFS ディスパッチャを構成の要とし、既存の NFS クライアントおよび NAS に依存することなく導入可能であることを特長のひとつとしている。

3．仮想化方式の動作原理

3.1. システムの構成

図1に見るように、NFS の要求と応答は、全て NFS ディスパッチャを経由する（in-band）構成となっている。NFS ディスパッチャは、その起動時に各 NAS がエキスポートしているファイルシステムのマウントを実行し、マウントポイントのファイルハンドルを獲得する。そして、それらのファイルハンドルを基に、新たに NFS クライアントへのエキスポートを設定後、これを NFS クライアントにマウントさせることで起動時の作業を完了する。

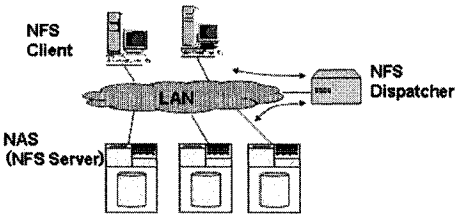

図1：システム構成

3.2. NFS のファイルハンドル処理

NFS 環境では、NFS クライアントの特定の要求に対し、NAS からファイルハンドルを戻り値として NFS クライアントに返す場合がある。ファイルハンドルとは、ファイルオブジェクト（ファイル、ディレクトリを指す）を NFS サーバで特定するための ID であり、プロトコル上規定されているのは、そのデータ長のみである。したがって、そのデータは各 NAS が管理しやすいように独自に生成され、NFS クライアントはデータの内容を一切解釈しないことが特徴となっている。

NFS ディスパッチャでは、この性質に着目し、戻り値に含ま

---

* A Study on Virtualization for File Servers
Satoshi Yamakawa, Wataru Katsurashima, Jun Ishikawa, and Yoshihide Kikuchi
Internet Systems Research Laboratories, NEC Corporation.

れる全てのファイルハンドルを、NAS のどのエキスポートからのものであるかが識別できるように変換し、NFS クライアントへ応答を返す。これにより、以降、NFS クライアントは変換されたファイルハンドルを用いた NFS 要求を必ず発行するため、NFS ディスパッチャではファイルハンドルを確認するだけで、どの NAS へ要求を転送すればよいのかを瞬時に判断することが可能となる。

この NFS ディスパッチャでの、NFS クライアント側と NAS 側におけるファイルハンドルの組み替え処理は、次節に述べる単一名前空間を構成するための NFS プロトコル転送の基本アルゴリズムとなっている。

3.3. NFS ディスパッチャによる名前空間の構成方法

複数の NAS のファイルシステムイメージを仮想化して、NFS クライアントに提供するためには、NAS 間を横断する統一的な名前空間を構成する必要がある。

図2は仮想化におけるディレクトリツリーの構成例を示したものである。これは、NAS_1 のディレクトリ A のサブディレクトリとして存在しているディレクトリ B に関するアクセスを、NFS クライアント透過に NAS_2 のディレクトリ C に関するアクセスとして処理することを表している。このようにファイルシステムをまたがって形成されたディレクトリツリー部分を以降ファイルシステム分岐点（FS 分岐点）と呼び、NFS ディスパッチャにより、従来のマウントポイントの状態を拡張したポリシーに従って処理が行われる。これにより、NAS ごとに管理されている名前空間を、NFS クライアントに対しては、複数 NAS を横断した単一名前空間として統合して見せることが可能となる。

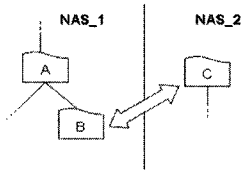

図 2：名前空間の統合の構成例（FS 分岐点）

3.4. FS 分岐点での処理

NFS ディスパッチャでは、図2のような FS 分岐点へのファイルアクセス要求を処理するために、通常のファイルハンドルの組み替え処理を拡張したアルゴリズムが実行される。NFS ディスパッチャは、全ての FS 分岐点のファイルハンドルを管理情報として持ち、ファイルアクセス要求に付加されているファイルハンドルが FS 分岐点のものであり、かつ単純転送では所望の応答が期待できない要求の場合に拡張処理を実施する。この拡張処理は、例えば、NFS ディスパッチャが複数の NAS へ要求を送り、戻ってきた複数の応答情報を再構成して NFS クライアントへ適切な応答を返すといったものである。

3.5. NFS の要求、応答の処理フロー

図 3 は、以上をまとめ、NFS ディスパッチャでの一般的な要求、および応答の処理フローを表したものである。

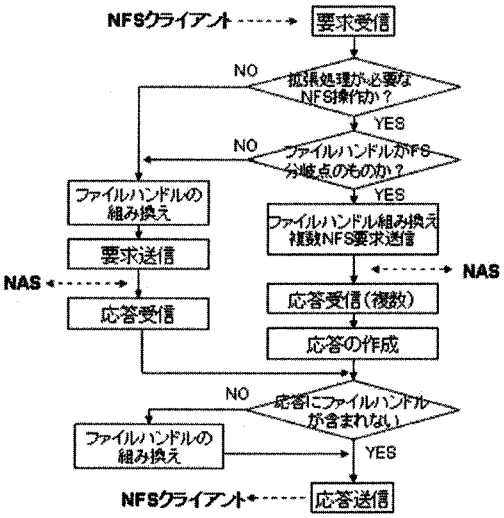

図 3：NFS ディスパッチャにおける処理フロー

4．NFS ディスパッチャの利点

FS 分岐点の作成およびその処理により、物理的な NAS の構成を NFS クライアントに意識させずに、すなわち、NFS クライアント見えの単一名前空間を変更することなく、容量逼迫時などのリソース再配置や新規 NAS の追加ができることが、NFS ディスパッチャ導入による大きなメリットといえる。

また、In-band 機器では、全ての要求、応答が NFS ディスパッチャに集中して性能のボトルネックとなりやすいという問題も、軽量なファイルハンドル組み替え処理により解消されていることも特長である。すなわち、Web スイッチのようなアプリケーションレイヤのスイッチと比べて巨大なテーブルの検索の必要もないことから、非常に高速な実装が実現できる。

5．むすび

本稿では、複数の NAS を仮想化、統合する NFS ディスパッチャを提案した。今後は、NFS ディスパッチャの試作をすすめ、性能評価などを行っていく予定である。

参考文献

[1] J.S. Chase, “Server switching: yesterday and tomorrow,” Proc. of the 2nd IEEE Workshop on Internet Applications, pp.114–123, 2001.

[2] R. Passmore, “Asymmetrical Virtualization at Last: TrueSAN’s Paladin,” Gartner Note Number: T–14–1374, 2001.

[3] Darrell Anderson, et al., “Interposed Request Routing for Scalable Network Storage,” Proc. of the 4th Symposium on Operating System Design & Implementation, 2000.

[4] Tricord Systems Inc., “Decentralized file mapping in a striped network file system in a distributed computing environment,” US Patent 6,029,168.

[5] Sun Microsystems, Inc., “NFS: Network File System Protocol Specification,” IETF, RFC1094, 1989.

[6] B. Callaghan, B. Pawlowski, P. Staubach, “NFS Version 3 Protocol Specification,” IETF, RFC1813, 1995.