ファイルサーバ仮想化アプライアンス NAS スイッチの提案

桂島 航　　石川 潤
NEC インターネットシステム研究所

概要

近年、情報の多様化、大容量化を追い風として、NAS (Network Attached Storage) が急速に普及している。しかし、NASは各々が独立したサーバとして稼働するため、ノード数が増えても容量や性能がスケーラブルに向上しないうえに、さらに管理は複雑になるという課題を抱えている。本論文では、上記課題を解決するNAS仮想化アプライアンスの一構成法を提案する。提案するアプライアンスを使うことで、デファクトスタンダードであるCIFSプロトコルをクライアントとNAS双方が使用する環境であれば、既存のクライアントとNASには全く手を入れずに、複数台のNASを連携させてスケーラブルな容量と性能の向上を実現することができる。

The NAS Switch:　An Appliance to Virtualize File Servers

Wataru Katsurashima　　Jun Ishikawa
Internet Systems Research Laboratories, NEC Corp.

Abstract

Continuously growing and diversifying data is rapidly accelerating the spread of NAS (Network Attached Storage). However, increasing the number of NASes does not improve the performance and capacity of system, and merely complicates the management problem because each NAS behaves as an independent server. To solve the problems, this article proposes an appliance to virtualize NASes. The appliance can aggregate NASes and provide scalable capacity and performance without modifying existing clients and NASes that both use CIFS protocol of a de facto standard.

1.　序論

近年、データセンターやオフィスのワークグループで利用されるストレージとして、NAS（Network Attached Storage）と呼ばれるファイルサーバ・アプライアンスが急速に普及している。NASが社会に広く受け入れられた一因は、その導入の容易さにある。すなわち、NASは、プロトコル互換性の検査や新たなネットワーク導入などの下準備なしに、ほぼすべての既存クライアントが利用することができるという利点を有している[1]。

しかし、NASは、最初に導入した1台だけでは容量が足りなくなり、2台、3台とNASの数が増えるに従って、管理作業が複雑化してくるという問題点を抱え

[1] NFS（Network File System）、CIFS（Common Internet File System）といった主なファイル・アクセス・プロトコルは、事実上の標準となってから多くの年月が経ち、ほとんどのプラットフォームでサポートされている。また、これらはIPネットワーク上で利用することができる。

ている。これは、個々のNASは独立したサーバとして機能するために、ユーザや管理者は複数のNASに対して、個々に設定／管理を行う必要があるためである。また、あるNASの空容量がひっ迫し、管理者がNAS間でデータ・マイグレーションを行う必要が生じた場合には、全てのクライアントにその結果を反映させるための再設定が必要となる点も改善が望まれている。このような問題に対して、大型のNASで従来のNASをリプレースするという対処法もあるが、大型のNASは購入に際して初期投資が大きいうえに、この先容量不足にならないという保障もない。

以上のように、スケーラビリティに関する問題を既存のNASは抱えており、今後、大規模なNASシステムが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）

ことが必須であると考えられている。

これまで、これらの問題を解決するためのアプローチとして、大別してクラスタリング、Out-of-band型仮想化の二種類の方法が提案されている。

一つ目のクラスタリングとは、各NASのファイルシステム間でクラスタ・システムを構築し、NAS全体の容量と性能のシームレスな向上を図るというものである[1][2]。しかし、この方法では、全てのNASを同一の専用アーキテクチャで揃えるか、全てのNASに専用ソフトウェアをインストールする必要がある。そのため、既存NASをクラスタに加えることができない、あるいは専用OSにて高性能化を図っているNASへはソフトのインストールができないなどの問題点があった。また、規模が大きくなるにつれてNAS間での通信量が増大するため、スケーラビリティにも限界があると考えられる。

二つ目のOut-of-band型仮想化とは、ディレクトリツリーとファイルの位置情報のマッピングを、ストレージからは独立したメタデータサーバに管理させることで、システム全体としての名前空間を形成する方法である[3][4][5][6]。これらの方法では、クライアントのOSに組み込まれたリダイレクタが、メタデータサーバに問い合わせることで共有ディレクトリやファイルの位置情報を入手し、アクセスを行う仕組みとなっている。しかし、この方法は、クライアントとNAS双方で専用のプロトコルを使用しなければならないか[3][4]、もしくはクライアントに専用ソフトウェアをインストールした上で、SAN（Storage Area Network）のような特殊なネットワークを別個に構成しなければならない[5][6]という運用上の難しさを課題として抱えている。また、SANを導入した場合、既存のNASを継続して使用することができないという問題点もあった。

以上のように、これまでのアプローチの多くは、クライアントもしくはNASが特殊な仕様を満たすように規定されるため、既存環境との融和性、将来の拡張性に課題を抱えていた。

本論文では、このような課題に対して、既存のクライアントとNASには全く手を入れずに、複数台のNASを連携させ、スケーラブルな容量と性能の向上（スケールアウト）を実現する一方式を提案する（以下では、NASを対象に議論を進めるが、提案手法はそのまま汎用のファイルサーバに適用可能である）。提案する方式は、クライアントとNASの通信路の中間に位置するIn-band型アプライアンスにより実現される。本方式では、クライアントとNAS双方が事実上の標準ファイル・アクセス・プロトコルの一つであるCIFSを使用する環境において、複数の異種NAS同士の名前空間を統合し、新規NASの追加やリソースの再配置などをユーザから透過に実行することができ、NAS間の連携や大規模システムの管理を容易にすることが可能となる。なお、筆者らはもう一つの標準であるNFS環境における方式についても提案を行っているので参照されたい[7]。

以下、2章で既存技術の代表であるDFSについて概要を説明し、3章で提案するNASスイッチの説明を行い、4章で提案手法に対する考察を加える。

2. 従来技術 DFS の概要

本章では、提案する NAS スイッチと同様の機能を提供する Out-of-band 型仮想化の従来技術である DFS (Distributed File System)[4] について、その課題も交え概観する。また、NAS スイッチが活用する DFS のクライアント側の機能であるリダイレクト機能についても、提案手法を理解するための準備として説明を加える。

2.1. DFS の概要とその課題

CIFSプロトコルは、DFSと呼ばれる仮想名前空間を構成する仕組みを備えている。DFSを使えば、複数のNASが公開する共有フォルダを、一つの共

有フォルダのサブフォルダとして統合し、CIFSクライアントに仮想名前空間を提供することができる(図1)。これらのサブフォルダは、共有フォルダから他の共有フォルダへのリンクとして機能し、クライアントが各共有フォルダの名前空間を渡り歩くのを助ける。仮想名前空間を利用することにより、クライアント透過に、NASの追加やNAS間のデータ・マイグレーションが可能になる。

しかし、DFSは既存環境との融和性や、拡張性という点で、いくつかの課題を抱えている。たとえば、DFSのサポート状況であるが、クライアント側はほとんどがサポート済みであるのに対して[2]、NAS側ではサポートされていない機種も少なくないというのが現状である。また、CIFSと密接に連携した技術であるためにNFSクライアントに対応することができないという制限もあった。さらに、仮想名前空間の構成は、リンクを張ることができる共有の数がNAS1台あたり1つに限られるために階層化できる自由度が低く、実際には図1のようなフラットな構成で利用されることが多かった。

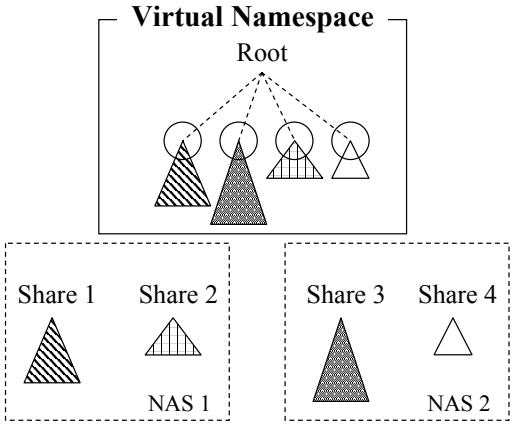

図1　DFS の仮想名前空間の構成例

以上の課題をまとめると

1. 既存環境と連携できない場合がある
   - DFSに対応していないNASとは連携できない
   - NFS／CIFSの両方のクライアントから同じ仮想名前空間を使用することができない
2. 仮想名前空間の構成がスケーラビリティ、フレキシビリティに乏しい
   - 他の共有にリンクを張ることができる共有、すなわち共有のルートの数はNAS1台あたり1つに制限される

となるだろう(3章では、NASスイッチが、これらの課題をどのように解決するかを詳細に議論する)。

2.2. DFS の動作とクライアントの機能

我々が提案するNASスイッチは、DFSのクライアント側の機能であるリダイレクト機能を活用している。そこで、本節では、提案手法を理解するための準備として、DFSが仮想名前空間をユーザに提供する際の動作シーケンスについて説明する。

DFS では、各 NAS は、自身が管理する名前空間と、他の NAS が管理する名前空間とをつなぎ合わせるために、そのつなぎ目に関する情報を持っている。以下、このつなぎ目のことを共有のつなぎ目と呼ぶことにする。DFS をサポートする NAS は、クライアントから受信した要求が共有のつなぎ目をまたぐパス名を含むものであれば、クライアントにエラーを返す。クライアントは、エラーを受けると、GET_DFS_REFERRAL というリソースの正しい接続先、すなわち照会先(referral)を得るための要求(以下、照会先の発行要求)を NAS に送る。NAS は、照会先として NAS のコンピュータ名、共有名、パス名を含む、要求が指定するリソースの場所に関する情報をクライアントに返す。これにより、クライアントは問合せをすべき正しい照会先を受け取り、ここに当初の要求をリダイレクトする。以上のやり取りは、ユーザからは透過に行われるので、ユーザはシームレスに NAS 間の名前空間を渡り歩くことができる。次章に述べるように、このクライアントのリダイレクト機能を NAS スイッチも利用している。

3. NAS スイッチ

提案する方式は、NAS スイッチというクライアントと NAS の間に配置される In-band 型アプライアンスを用いて実現される。本章では、まず提案するシステムの構成について説明したあと、NAS スイッチが実現する仮想名前空間の提供機能についてまとめ、

[2]ここではスタンドアロン DFS を想定している(CIFS クライアントは、Windows OS だと、Windows98 以降が対応している)。

ついでその機能がどのように実現されるのかを詳細に述べる。

3.1. 提案するアーキテクチャ

NASスイッチは、既存のクライアントとNASには全く手を入れずに、複数台のNASを連携させ、スケーラブルな容量と性能の向上（スケールアウト）を実現することを目的としている。この目的の実現のために、我々が提案する基本アーキテクチャを図2に示す。クライアントとNAS双方が標準ファイル・アクセス・プロトコルの一つであるCIFSを使用する環境において、NASスイッチが単純にIPネットワークに接続されているだけの構成である。ここで、クライアントとNASに専用ソフトウェアなどのインストールを行う必要はなく、かつ、NASはそれぞれ完全に独立して動作し、同期通信などのオーバーヘッドも発生しないことに注目されたい。

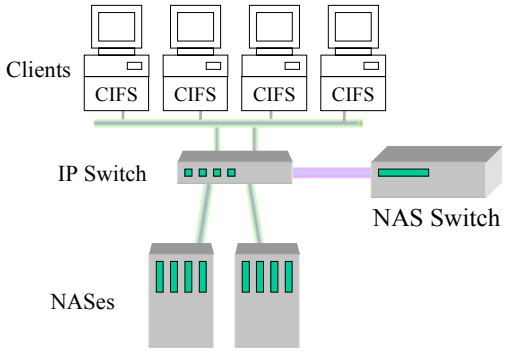

図2 NASスイッチを用いたシステム構成

3.2. NASスイッチの機能

NASスイッチは、複数台のNASを連携させた仮想名前空間をユーザに提供する機能を持つ。DFSと比較した場合の利点は

1. 既存環境と連携できる
   - DFSに対応していないNASの共有からも他の共有へリンクを張ることができる
   - NFS／CIFS両方のクライアントから同じ仮想名前空間を使用することができる
2. 仮想名前空間の構成にスケーラビリティ、フレキシビリティがある
   - NAS1台あたりで、他の共有にリンクを張ることができる共有の数に制限がない
   - 共有のルートからでなくても、他の共有にリンクを張ることができる（階層化構成に制限がない）

である。ただし、NFS／CIFS両方のクライアントに同じ仮想名前空間を提供する機能は現在検討中の課題であり、その内容については4章で議論する。

図3は、NASスイッチが提供する仮想名前空間の構成例であり、以下、次節では、このような機能がどのように達成されるかについて述べる。

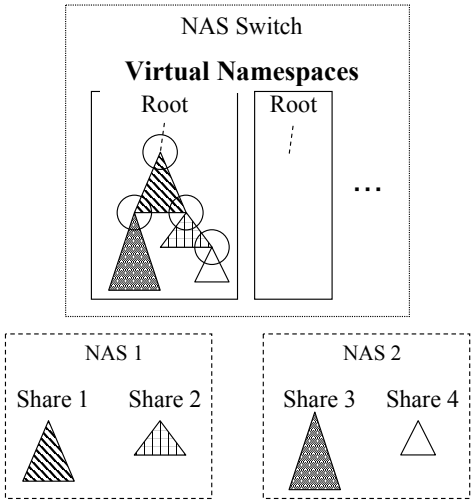

図3 NASスイッチの仮想名前空間

3.3. NASスイッチの構成

NASスイッチは、バックエンドの各NASを、NASスイッチ自身が提供する仮想NASとしてクライアントに見せている。クライアントはこの仮想NASに接続するので、NASスイッチは、クライアントとNASの通信を常時監視することができ、必要に応じてクライアントからのパケットに自身で応答したり、あるいは、それを仮想NASに対応するバックエンドの実際のNASにフォワードしたりする。これがNASスイッチの基本アーキテクチャである。この構成では、NASスイッチは、CIFSクライアントのDFS機能と連携して動作するため、クライアント側は2.2節で述べたDFSに対応していなければならない。しかし、CIFSクライアントは、ほぼDFSに対応しているので、特殊なソフトウェアをインストールする必要はない。

以上に述べたNASスイッチの構成を具体的にブロック図としたものが図4である。この図にみるように、NASスイッチは、大きく分けて、インターセプト判断手段、照会先発行手段、つなぎ目情報テーブルから構成される。以下、まず、各手段の詳細について説明する。

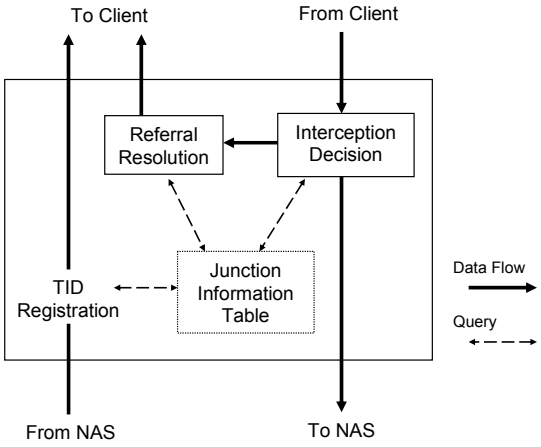

図4　NASスイッチの構成

3.3.1.　インターセプト判断手段

インターセプト判断手段とは、クライアントから仮想NASに対する要求を受信した場合に、インターセプトが必要な要求であれば、要求をNASに転送せずインターセプトし、そうでない要求はバックエンドのNASにフォワードする手段である。インターセプトした手段は、後述の照会先発行手段に送られる。

インターセプトする要求は以下のふたつである。

- 共有のつなぎ目をまたぐ要求
- 照会先の発行要求

共有のつなぎ目をまたぐ要求かどうかは、要求の種類と、要求が指定するオブジェクトをチェックすることで識別する。ファイル・アクセス・プロトコルにおける要求の種類は、READやWRITEのようなファイル操作系と、OPENやRENAMEのようなディレクトリ操作系に大別できるが、共有のつなぎ目をまたぐ可能性のある要求はディレクトリ操作系である。要求がディレクトリ操作系である場合には、インターセプト判断手段は要求が指定するオブジェクトを、要求が使用しているセッション、要求に含まれるTID（共有を特定するID）、パス名から特定する。特定したオブジェクトが、クライアントが現在接続している共有と異なる共有に存在する場合には、インターセプト判断手段は要求が共有のつなぎ目をまたぐ要求であると判断する。

照会先の発行要求かどうかは、要求の種類がGET_DFS_REFERRALであるかどうかをチェックすることで判断できる。

3.3.2.　照会先発行手段

照会先発行手段は、インターセプト判断手段から要求を受信すると、以下のようにクライアントに適切な応答を発行する。共有のつなぎ目をまたぐ要求である場合には、PATH_NOT_COVEREDというエラーコードをクライアントに発行する。また、照会先の発行要求である場合には、要求が指定するオブジェクトが存在する NASに対応する仮想NASのコンピュータ名、共有名、パス名を、照会先としてクライアントに発行する。

3.3.3.　つなぎ目情報テーブル

つなぎ目情報テーブルは、共有のつなぎ目に関する情報を格納している。この情報は、基本的には、管理者が前もって形成した仮想名前空間を構成する、各名前空間の接続ポイントの情報がベースとなって作成される静的な情報である。

ただし、クライアントは要求の中で共有を指定する際に、NASが発行する揮発的なIDであるTIDを用いるので、TIDと共有名の対応付けは常時更新する必要がある。具体的には、共有への最初の接続要求である Tree_Connect の応答でTIDが発行されるので、その応答を拾ってテーブルに記載しておく。TIDはセッションが切れると無効になるので、その際にエントリを消去する。

3.4.　NAS スイッチの動作

本節では、最後にクライアントがNAS1の共有Aに接続してからNAS2の共有Bにリダイレクトするまでの一連の流れを例にとり、NASスイッチが、前節で説明した各手段により、どのように仮想統一名前空間を提供するかについて説明する。

いま、NASスイッチでは、共有Aの */home/user1* が共有Bの */data* にリンクするように設定しているとする。さらに、NAS1は仮想NAS01、NAS2は仮想NAS02にそれぞれ対応しているとする。この場合、以下のような動作が実行される。

1. クライアントは、仮想NAS1の共有Aに */home/user1/file* のOPEN要求を送る
2. NASスイッチは、セッションと、要求に含まれるTIDから要求が共有Aに対するものであると、さらに*/home/user1* が共有のつなぎ目をまたがると判断して要求をインターセプトし、クライアントにPath_not_coveredエラーコードを発行する
3. クライアントは、仮想NAS1の共有Aに

/home/user1/file の照会先発行要求を送る

4. NASスイッチは、コマンド番号から照会先の発行要求であると判断してインターセプトし、クライアントに照会先として仮想NAS02、共有B、*/data/file* を発行する
5. クライアントは仮想NAS02の共有Bに */data/file* のOPEN要求を送る

以上の動作は、NASの特定の機能などを利用することなく行われるので、NASがDFSに対応している必要はなく、また共有のルートからでなくとも自由に他の共有へリンクがあるようにクライアントに仮想的な名前空間を見せることができる。

NASスイッチは、以上のように共有のつなぎ目部分をNASからは独立して設定して、クライアントに仮想名前空間を提供する。また、自身で要求を処理するのは、その共有のつなぎ目部分に関する部分だけであり、それ以外は各NASにフォワードするため実装が軽いという特長も有している。

4. 考察

本章では、これまでに提案されてきたクラスタリング、Out-of-band型仮想化と比較した場合の、In-band型仮想化アプライアンスであるNASスイッチの長所、短所について議論し、その中で今後の研究についてもふれる。

In-band型の長所は、序論でも述べたが、クライアント及びNASに、特殊なソフトウェアをインストールする必要がないことにある。筆者らは、NFS版においてもIn-band型の仮想化方法を開発しており[7]、将来的にはこの方法と統合を行うことにより、ほぼ全てのクライアントとNASが、特殊なソフトウェアのインストールをすることなく、仮想名前空間を利用することができるようにする予定である。また、In-band型には、システムに新たな機能を追加することが容易であるという長所もある。たとえば、今まではNAS全てに新たな機能を導入する必要があったのが、NASスイッチだけに新たな機能を導入するだけで良くなるということである。追加する機能としては、セキュリティ機能の強化などが考えられる。

次に、In-band型の短所であるが、ボトルネックとなる可能性があること、単一障害点となる可能性があること、レイテンシが増加することが挙げられる。最初の二点に関しては、NASスイッチをN+1冗長化することで、NAS群の性能を限界まで引き出し、フォールトトレラントなシステムとすることが可能であり、これも今後の課題と考えている。レイテンシに関しては、NASスイッチが共有のつなぎ目と関係ないパケットをどれだけ高速に処理することができるかがポイントになる。インターセプト判断手段をカーネル内に作り込むなどの実装面での工夫や、インターセプト判断手段内での判定条件を工夫するなどの方法が考えられる。

5. 結論

本論文では、クライアントとNAS双方が標準ネットワーク・ファイル・アクセス・プロトコルの一つであるCIFSを使用する環境において、既存のクライアントとNASには全く手を入れずに、複数台のNASを連携させ、スケーラブルな容量と性能の向上(スケールアウト)を実現する一方式を提案した。

参考文献

[1] Tricord Systems, Inc. Lunar Flare NAS http://www.tricord.com/appliance/

[2] 1Vision Software, Inc. vNAS http://www.the1vision.com/products/vnas/

[3] Andrew File System http://www-2.cs.cmu.edu/afs/andrew.cmu.edu/usr/shadow/www/afs.html

[4] Distributed File System http://www.microsoft.com/windows2000/techinfo/howitworks/fileandprint/dfsnew.asp

[5] Anupam Bhide, et al, "File Virtualization with DirectNFS, " Proc. of the 10th NASA Goddard Conference on Mass Storage Systems and the 9th IEEE Symposium on Mass Storage Systems, pp. 43-58, Maryland, USA, 2002.

[6] EMC, Celerra HighRoad http://www.emc.com/products/software/highroad.jsp

[7] 山川聡, 桂島航, 石川潤, 菊地芳秀, "ファイルサーバの仮想化方式に関する一検討", 情処全大 Vol.64, No.3, pp501-502, 2002.