ファイルサーバ向け仮想化機能の設計と実装（1）

中野　隆裕†　松沢　敬一†　揚妻　匡邦†　亀井　仁志†
（株）日立製作所　システム開発研究所

1．はじめに

計算機の処理性能向上と仮想化技術の発達により、従来複数のサーバで行っていた処理を、少数サーバに集約することが可能となってきた。

ファイルサーバにおいても、複数の古いファイルサーバの置き換えや、部署ごとのファイルサーバを統合するなど、集約の応用が考えられる。装置や管理対象を集約することで、運用コストに加え電力消費量削減の効果も期待できる。

我々は、上記のようなファイルサーバの集約を提供する仮想化方式VNAS（A Virtualization method for NAS）を開発し、Linux カーネル上に実装した。

2．ファイルサーバ仮想化のメリット

ファイルサーバを集約する際、単にデータを集約するだけであれば、新たなファイルサーバにデータを移行するだけで済む。しかし、集約する古いファイルサーバ毎にアカウント管理が異なる場合や、サーバ名や共有名変更に伴うクライアント側の設定変更が膨大となる場合には、ユーザアカウントの付け替えや、全クライアントの設定変更などの作業が必要となる。

ファイルサーバ仮想化により、仮想ファイルサーバが利用可能であれば、移行前のファイルサーバ毎に仮想ファイルサーバを割り当て、アカウント情報、ホスト名、共有名などを継承した上で、データ移行することで、ユーザアカウントの付け替えや、クライアントの設定変更なく、ファイルサーバの集約が容易に実現できる。

3．ファイルサーバ仮想化の課題

ファイルサーバ仮想化は、仮想化技術を利用する。仮想化技術には、(1)HW仮想化方式、(2)ハイパーバイザ方式、(3)OSレベル仮想化方式など様々な方式がある[1]。

これら方式には、表1のような差異がある。

方式(1)(2)は、異種OSを同時稼動できるが、入出力オーバヘッドが大きい。方式(3)は、異種OSには対応出来ないが、入出力はOS本来の機能で行えるため、低オーバヘッドで処理できる。

表 1　仮想化方式による差異

方式	(1)	(2)	(3)
異種OS対応	○ 広く対応	△ OS対応要	× 同一のみ
仮想化レベル	高 HW全機能	中 CPU,メモリ	低 参照情報
入出力性能	× エミュレーション要	△ メモリコピー要	○ 追加オーバヘッドなし

ファイルサーバを集約する目的では、複数種類のOSを稼動させる必要はなく、入出力オーバヘッドを低く抑え、効率よく仮想ファイルサーバを稼動させられる特性が重要となる。

さらに、方式(3)には、方式(1)(2)と異なり、OS の機能を部分的に仮想化する実装が可能である。このアプローチは、完全な仮想化を提供するものではないが、特定用途の使用で機能範囲が限定可能な場合には、仮想化機能の開発量を抑えられ、実現を容易化できる。

本稿では、方式(3)を前提に、仮想化すべき機能範囲の明確化、および、仮想ファイルサーバ間での機能の相互干渉を防止するための基本設計について述べる。

4．VNAS 設計

VNAS の設計の要件は、複数の仮想ファイルサーバが互いに干渉することなく正しく動作できることである。

ここでは、まず、VNAS の基本構造を示す。次に、ファイルサーバが利用する下記 OS 機能での干渉排除方式を示すことで、上記の達成を示す。

(a) ネットワーク通信
(b) ファイル入出力
(c) 設定情報参照（ユーザ管理含む）
(d) プロセス間通信

4.1 基本構造設計

VNAS は、仮想サーバ(Vsv)にてファイルサービスを行い、物理サーバ(Psv)にてリソース管理や仮想サーバの制御を行う。

A Design and Implementation of the Linux Virtualization for File Server
† Systems Development Laboratory, Hitachi, Ltd.

Psv は、装置全体の管理者によって操作され、ハードウェア管理、障害処理、クラスタリング制御に加え、Vsv の設定やリソース割当（IP アドレスやブロックデバイス）などを行う。Vsv では、Psv から割り当てられた IP アドレスとブロックデバイスを用いて、ファイルサービスを提供する。

VNAS では、OS レベルで仮想化を行うため、Vsv 毎に OS 内部データの一部（マウントテーブルなど）を個別に持つ。

この様に OS 内部データを、全 Vsv で共有するものと、Vsv 毎に個別に持つものに分割することにより、図 1 に示すようなファイルサーバ集約が可能となる。

図 1 の例では、Vsv1 は部門 1 用の仮想ファイルサーバで、Vsv1 クライアントからアクセスされる。Vsv1 は、部門 1 の LDAP サーバを用いてユーザ管理している。同様に、Vsv2 は部門 2 用の、Vsv3 は部門 3 用の仮想ファイルサーバで、Vsv2 が NIS、Vsv3 が passwd ファイルを用いてユーザ管理している。

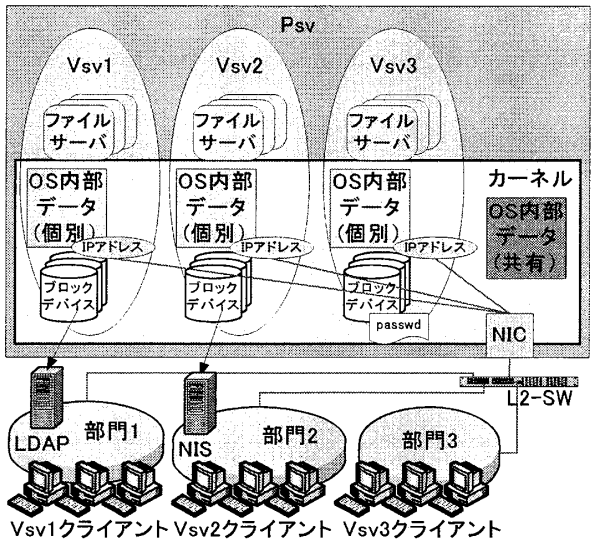

図 1　VNAS 構成例

4.2 干渉排除方式設計

4.2.1 ネットワーク通信

ネットワーク入出力の仮想化は、Psv から割り当てられた IP アドレスによって Vsv を区別する。ファイルサーバは、Psv から割り当てられた IP アドレスで待ち受けることで、クライアントからの要求を受け取ることができる。

Psv から割り当てられていない IP アドレスでの待ち受け要求は、他の Vsv と干渉を防ぐため、カーネルがエラーを返す。

4.2.2 ファイル入出力

ファイル入出力の仮想化は、ブロックデバイスの分割（Psv による割当て）とファイル名前空間の独立化で行なう。ブロックデバイスは、Psv から割り当てる際に、他の Vsv と干渉しないよう分割する。ファイル名前空間は、Vsv 毎の OS 内部データにあるマウントテーブルによって独立化され、さらに、ルートファイルシステムも個別に割り当てることで、他の Vsv とファイルシステムとの干渉を防ぐ。

また、Psv から割り当てられていないブロックデバイスのアクセスは、他の Vsv と干渉を防ぐため、カーネルがエラーを返す。

4.2.3 設定情報参照（ユーザ管理含む）

ユーザ管理を含む設定情報などの設定ファイルは、どの Vsv でも同じパスに存在するが、Vsv 毎に異なるルートファイルシステムに格納することで、他の Vsv との干渉を防いでいる。

4.2.4 プロセス間通信

プロセス間通信は様々な方法があるが、シグナルを除き、ファイル名前空間と連携しており、4.2.2 で示したとおり、他の Vsv と干渉しない。シグナルは、他の Vsv との干渉を防ぐため、他の Vsv を指定した要求にはエラーを返す。

以上の干渉排除方式により、ファイルサーバとして機能させるために必要充分な仮想化機能を実現している。

5. おわりに

複数のファイルサーバを実行可能にする仮想化方式 VNAS を設計、実装し、その動作を確認した。さらに、本方式の性能評価[3]、クラスタ構成への拡張[4]を行っている。

6. 参考文献

[1] 高橋雅彦，“Linux Containers/Cgroups BoF: サーバから組込みまで”,Japan LINUX Conference 2008.

[2] B. Clark, “Xen and the Art of Repeated Research”, USENIX 2004 Annual Technical Conference, pp. 135-144, 2004.

[3] 松沢敬一，“ファイルサーバ向け仮想化機能の設計と実装（2）”,情処 71 回全国大会.

[4] 亀井仁志，“ファイルサーバ向け仮想化機能の設計と実装（3）”,情処 71 回全国大会.