ファイルサーバ向け仮想化機能の設計と実装（2）

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

1. はじめに

近年計算機の処理性能が向上し，従来複数のサーバで行っていた処理を少数のサーバで賄えるようになった．そこで，数々の仮想化技術が開発されサーバ集約に活用されている．

我々はファイルサーバ集約のための仮想化方式VNAS(A Virtualization method for NAS)を開発し，Linux カーネル上に実装した．本実装では，カーネル内変数の仮想サーバ毎分割と，資源の仮想サーバタグ付け方式により，大量の I/O を処理できる低オーバヘッド性を保ち，かつ仮想サーバごとの十分な独立性を保った仮想化を実現できた．

2. 従来の仮想化方式の課題

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]．

ファイルサーバはネットワーク上の PC からアクセス要求を受けてファイル入出力を行うため，ディスクや NIC への I/O を多用し，また多数のプロセスが動作しタスクスイッチも頻発する．よって I/O 時にホスト・ゲスト環境間でデータコピーが発生したり，環境切り替えに CPU 上のキャッシュパージやレジスタ入れ替えが発生する方式(1)や(2)のファイルサーバへの適用は性能が問題となる．

ファイルサーバにおいては，ファイルサービス上に現れる資源，例えばファイル名前空間や IP アドレスが仮想サーバ間で独立でなければならない．一方，ファイルサーバの利用者が直接参照しない資源は独立である必要はない．例えばメモリ管理を仮想サーバ毎に独立に持たせず，サーバ全体で共有することで，負荷の高い仮想サーバが多くのメモリを利用することができる．方式(1)(2)ではメモリも各仮想サーバに固定で割り当ててしまう．この様に資源分割の粒度が課題となる．

また，ファイルサーバ上では割り込みや workqueue の様にプロセスコンテキスト外で行われる処理も，仮想サーバ毎に処理できなければならない．方式(3)の既存の実装は各プロセスを仮想サーバに割り当てるため，プロセスコンテキスト外処理に対応できない．

このように，ファイルサーバ仮想化では，オーバヘッド，資源分割の粒度，プロセスコンテキスト外の処理の３つの課題を持つ．

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

3. 仮想化方式

VNAS は，OS レベル仮想化方式に，さらなるオーバヘッド削減とプロセスコンテキスト外処理への対応を加えた仮想化方式である．VNAS では仮想サーバ毎にプロセスや IP アドレスなどのファイルサーバ運用に必要な資源を分割し，変数のオフセット加算により仮想サーバ固有領域を参照可能にする．さらにプロセスなどの資源に識別タグ（以下，タグ）を付け分けることで各資源の他仮想サーバからの参照防止や，プロセスコンテキスト外の仮想サーバ遷移を正しく行う．我々はこの両方式により仮想化を実現する．

3.1 変数分割とオフセットによる低オーバヘッド仮想サーバ切り替え方式

我々は，ファイルサーバ運用に必要な資源がカーネル内変数に格納さいることに着目し，各変数を仮想サーバ毎に分割することで仮想サーバを構築できると考えた．

一般的な OS レベル仮想化は，Bharriprolu の方式[3]の様に資源を名前空間ごとに分け，ソースコード上で１段間接参照することで実現する．我々はその方式を発展させ，低オーバヘッドで仮想サーバを切り替えるため，単一のオフセット値変更で仮想サーバを切り替えるようにした．VNAS ではカーネル内の変数を仮想サーバ毎に分割する変数と，分割せず共有する変数の２つに分けた．分割の基準は，変数に対応する資源が仮想サーバごとに分かれる必要があるかどうかで決定する．例えばプロセス一覧情報や NFS の export 情報は分割し，メモリ管理機構やプロセススケジューリングに関する資源は全仮想サーバで共有し分割しない．

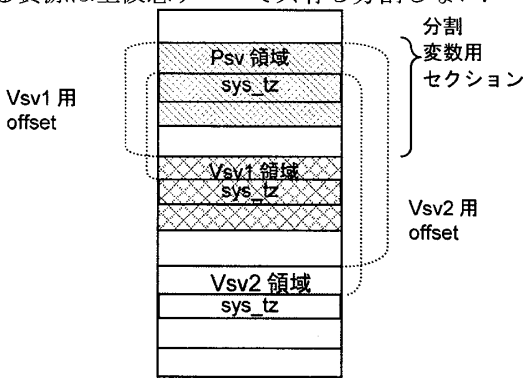

図 1 変数分割方式

カーネル内の変数は，リンカによりセクション単位でメモリ中の連続領域に配置される．VNAS で

はこの配置を利用し，分割変数を専用セクションに割り当て，分割変数だけを連続領域に集める．

運用中に仮想サーバ(Vsv)を新規作成する場合，カーネル内では分割変数群のセクションと同サイズのメモリを新規に割り当て，物理サーバ(Psv)用領域とのオフセットを計算する．カーネルが分割変数を参照する場合，元の Psv 用領域のアドレスにオフセットを加算したアドレスを参照することで仮想サーバ固有の変数領域を指定する．例えば，図１では変数 sys_tz に対し，オフセットを切り替えることで，仮想サーバ固有の sys_tz 格納領域を指定している．このオフセットは実行コンテキストの一部として各 CPU 固有メモリ領域及びプロセス管理構造体に格納される．そしてプロセススケジューリング契機で同時に仮想サーバが遷移する．

本手法により，各仮想サーバは固有のファイル名前空間や NFS export 情報，タイムゾーンなどを保持し，かつ低オーバヘッドで仮想サーバの切り替えが可能となる．

3.2 タグによる他サーバ操作防止と一時環境移行

本方式では，カーネルが持つ資源を表す構造体に，資源の所有環境を示すタグとして前述のオフセット値を付加する．対象の資源は，前述のプロセス以外に，タイマ処理，IP アドレスなどがある．

このタグは２つの目的で利用する．１つは，タグと異なる仮想サーバからの参照・変更防止である．例えば仮想サーバ１が設定した IP アドレスは，仮想サーバ２から変更させない．本手法は IP アドレス以外にシグナル送信先や，アクセスできるディスク，統計情報のアクセス管理などに用いる．

もうひとつの目的は，プロセスコンテキスト外の処理において，正しい仮想サーバで処理を行うための一時的な環境遷移に用いる．例えば仮想サーバ１が動作中に，タイマ割り込みで仮想サーバ２が登録した処理が行われると，仮想サーバ１の変数が上書きされデータ不正が発生する．このような不具合を防止するため，プロセスコンテキスト外の処理においては，一時的にタグに格納された環境に遷移して処理を行う．図 2においては仮想サーバ Vsv1 のプロセスが動作中にタイマ割り込みが発生すると，一時的にタイマ処理を登録したサーバに環境を切り替えて処理し，割り込みコンテキストが終了する前に元に戻る．

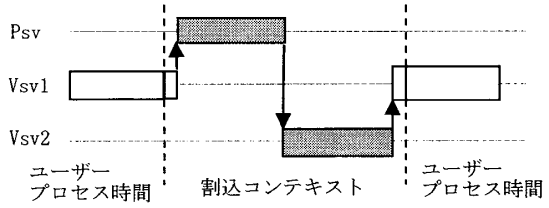

図 2 タグによる一時状態移行例

4. 評価

我々は Linux2.6 ベースカーネルの x86-64 アーキテクチャ用コード上に VNAS を実装した．そして物理サーバ上に固有の IP アドレス・ファイル空間を持つ複数の仮想サーバを立ち上げ，NFS・CIFS サービスが提供できることを確認した．

また，VNAS のオーバヘッドを評価するため，我々は iozone[4]を用いて NFS シーケンシャルアクセス時のスループットと CPU 負荷を測定した．

サーバには Xeon5140，4GB メモリを搭載する PC を利用し，RAID0+1 構成である 16 台の HDD 上に 4 つの XFS ボリュームを作成した．そして 4 台のクライアントから NFS 要求による負荷を与えた．

表 1に VNAS 実装前に対する測定環境毎のスループットと CPU 利用率の相対値を示す．VNAS 実装後の物理サーバの測定結果より，VNAS 実装のオーバヘッドが高々0.2%増であることを確認した．次に，仮想サーバ数を４まで増加して測定した場合，高々1.6%増となった．これらの測定より，VNAS のオーバヘッドは I/O 処理全体に対し小さいことが確認できた．

表 1 シーケンシャルアクセス性能相対値

測定環境	write			read		
	TP 比(%)	CPU 比(%)	CPU/TP 比(%)	TP 比(%)	CPU 比(%)	CPU/TP 比(%)
物理	+0.6	+0.8	+0.2	-0.1	-0.1	0.0
仮想×1	+1.1	+0.2	-0.9	0.0	+1.6	+1.6
仮想×2	0.0	+0.4	+0.4	0.0	+0.7	+0.7
仮想×4	+0.3	0.0	-0.3	-0.1	+1.5	+1.6

5. おわりに

本論文では，VNAS を Linux 上に実装し，低オーバヘッドで複数のファイルサーバが動作することを確認した．今後，仮想サーバ間のセキュリティ向上や，cgroups[1]などの研究成果取り込みによる計算資源の分割など，より柔軟な運用を行えるよう改良に取り組んでいく．

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] S. Bhattiprolu, "Virtual Servers and checkpoint/restart in mainstream Linux", ACM SIGOPS Operating Systems Review, Vol 42, Issue 5, pp.104-113, 2008.

[4] Iozone Filesystem Benchmark, http://www.iozone.org/ .