B_035

Xenのストレージ仮想化に関する考察
Storage Virtualization in Xen

藤田 智成† 稲垣 博人†
Tomonori Fujita Hirohito Inagaki

1. はじめに

近年，企業では，ITシステムの肥大化，複雑化に対応するために計算機が増加し，その運用，管理に必要な設備，人的リソースに伴うコストが大きな問題となっている．このため，複数の計算機のリソースを一台の計算機に統合する，仮想化技術に注目が集まっている．

複数の仮想計算機が一台の計算機で動作する仮想化環境では，計算機に接続された物理的なストレージデバイス（ディスク，テープ等）を仮想化し，各々の仮想計算機に専用のストレージデバイスが接続されているように認識させる必要がある．

本稿では，仮想化技術を実現するオープンのソースソフトウェアであるXen[1]が用いているストレージ仮想化方式に関して考察し，現在，我々が開発中の新たなストレージ仮想化方式について述べる．

2. Xenのストレージ仮想化方式

Xenのストレージ仮想化機構を図1に示した．Virtual Machine Monitor（VMM）は，ハードウェアの管理，複数の仮想計算機（ゲストOS）の管理を主に担当する．本稿は，ゲストOSとしてLinuxを想定する．

VMMの上では，ドメイン0と呼ばれる特別なゲストOSが1個と，ドメインUと呼ばれるその他のゲストOSが動作する．ドメイン0のゲストOSは，ドメインUのゲストOSと異なり，ハードウェアに直接アクセスする役割を持っている．VMMは，ドメイン0のゲストOSの機能を利用することで，ストレージの仮想化を実現する．ドメインUのゲストOSは，ドメイン0に接続し，ストレージへのI/O要求を発行し，その応答を受け取る．

以下では，現在利用可能な，2種類のストレージ仮想化方式について説明する．ストレージデバイスの種類はディスクとして議論を進める．

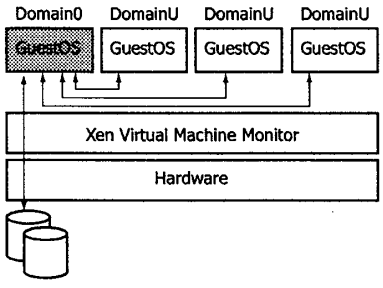

図1: Xenのストレージ仮想化

2.1 blkfront/blkback

blkfrontは，ドメインUのゲストOSで動作するデバイスドライバであり，ドメイン0のゲストOSで動作するblkbackデバイスドライバと接続し，ゲストOSから届いたディスクアクセス要求を伝える．blkfrontドライバによって，ドメインUのゲストOSは，ディスクが接続されているかのように認識する．blkfrontドライバは，ゲストOSに，SCSI，又はATAディスクとして認識させることが可能である．

ドメイン0で動作するLinuxは，通常の（仮想環境で動作していない）Linux同様に，物理ディスクを認識する．blkbackドライバは，blkfrontドライバの要求に応じて，物理ディスクに対するI/Oを実行する．

VMMは，ドメイン間の通信に使える共有メモリの仕組みを提供しており，ドメインUとドメイン0は，blkfrontドライバとblkbackドライバの通信のために，2つの共有メモリを利用する．blkfrontドライバは，1つ目の共有メモリに要求を順番に書き込み，2つ目の共有メモリから順番に応答を読み出す．blkbackドライバは，blkfrontドライバと逆の動作を行う．これらの共有メモリは，リングバッファ的に利用されている．

ディスクの読み出しでは，ドメインUのゲストOSが，ディスク位置，データ長，読み出したデータを保存するページフレームを，blkfrontドライバに通知する．blkfrontドライバは，1）指定されたページフレームへのドメイン0のアクセス許可をVMMに要求する，2）ページフレームのアドレス，ディスク位置，データ長を共有メモリに書き込み，ドメイン0に新しい要求の到着を通知する．ドメイン0のゲストOSのblkbackドライバは，1）指定されたページフレームをゲストOSのアドレスにマップし，2）物理ディスクから読み出したデータをページフレームに保存し，3）共有メモリに応答を書き込み，ドメインUに通知する．

ディスクの書き込みでは，blkbackドライバが，blkfrontドライバが指定したページフレームのデータを物理ディスクに書き込む点を除けば，読み出し処理と同様である．

blkbackドライバは，blkfrontドライバの要求に応じて，読み出し・書き込みを実行する対象として，物理ディスクに加えて，ソフトウェアRAIDのような仮想的なデバイスを利用することができる．通常ファイルを使う場合は，loopbackデバイス機能を使い，仮想的なデバイスとしてアクセスできる．

2.2 blkfront/blktap

blktapドライバは，blkbackドライバの代わりに使う目的で開発され，blkbackドライバ同様，blkfrontドライバと組み合わせて使用する．本稿では，現在開発が進められている，blktapドライバの新しい実装[2]について述べる．

blkbackドライバとblktapドライバの違いは，前者がドメイン0のゲストOSのカーネル内部でディスクI/O処理を実行するのに対し，後者はドメイン0のゲスト

†NTTサイバーソリューション研究所

OSのユーザ空間で動作するデーモンプロセスがシステムコールを使ってディスクI/O処理を実行する点である．

起動時に，blktapドライバは，制御用キャラクタデバイスを作成し，ユーザプロセス（tapdiskデーモン）との通信に利用する．tapdiskデーモンは，mmapシステムコールを使い，blktapドライバとblkfrontドライバの通信に使われる共有メモリを，ユーザ空間にマップする．tapdiskデーモンは，blkfrontドライバから新しい要求が届くまで，selectシステムコールを使いスリープする．

blktapドライバは，blkfrontドライバから新しい要求の通知を受け取ると，指定されたページフレームをゲストOSのアドレスにマップしてから，tapdiskデーモンを起床する．tapdiskデーモンは，mmapされたアドレス（共有メモリ）から新しい要求を読み，リード・ライトシステムコールを利用して，物理ディスクに対するI/Oを実行する．tapdiskデーモンは，制御用キャラクタデバイスに対するioctlを発行することで，I/Oが完了の完了をblktapドライバへ通知する．

2.3 考察

blkbackドライバは，ドメイン0のゲストOSのカーネル空間で全ての処理を実行するため，ユーザプロセスを使うオーバヘッド（システムコール等）を伴うblktapドライバと比較して，性能面で有利である．しかし，tapdiskデーモンが，リード・ライトシステムコールの代りに，ダイレクトI/Oと非同期I/Oインターフェイスを使って，ディスクI/Oを実行することで，blktapドライバは，blkbackドライバと同等の性能が実現できることが報告されている．

性能を別にすれば，blktapドライバは，ユーザプロセスを利用して，カーネル空間使うblkbackドライバよりも，様々な機能を容易に実装できるという利点を持つ．

そのような機能の例として，ファイルをそのままディスクイメージとして，ドメインUのゲストOSに提供するのではなく，専用のフォーマットを使い，ファイルにデータを保存することで実現するスナップショット機能がある．具体的には，セクタ等の単位で，データを間接参照するテーブルを持つフォーマットを利用し，変更があった領域は，データ保存のため間接参照先をファイルのどこかに割り当てる．割り当てられていないセクタは，スナップショット時のファイルを参照することで，簡単にスナップショットが実現できる．

二つ目の例として，ディスク毎のQoSやプロファイリング等の機能があげられる．複数の仮想計算機を動作させる仮想化環境では，この種のリソース制御のための機能は非常に重要である．

blktapドライバは，より高度なストレージ機能の実現を容易にしたが，物理デバイスと比較すると，その機能は，まだ不十分な点も多い．

例えば，SCSIデバイスと比較すると，1）動的にデバイスの追加・削除ができない，2）I/O命令の実行順序制御ができない，3）ディスク以外のデバイスの種類，テープ等に対応しない，等があげられる．

前述のように，blkfrontドライバによって，ドメインUのゲストOSには，SCSIディスクとして見えるが（ATAディスクとして見せることも可能），実際はSCSIプロトコルの機能を実現していないため，ユーザアプリケーションへの修正が必要になる場合もある．例えば，SCSI IDを使うアプリケーションは，全く動かない．

3. SCSIプロトコルを使うストレージ仮想化

2章で述べた課題を解決するためには，既存のblkbackやblktapドライバに機能を実装するアプローチが考えられる．しかし，我々は，大規模な実装を避け，既に提供されているゲストOSの機能を活用することで課題を解決する，SCSIプロトコルを使った，新たなストレージ仮想化方式の実装を進めている．

提案方式では，ドメインUで動作するドライバ（vscsifrontと呼ぶ）は，ゲストOSのSCSIホストバスアダプタドライバとして動作する．vscsifrontは，受け取ったSCSIコマンドを，ドメイン0で動作するvscsibackドライバに渡す．vscsifrontとvscsibackドライバは，blkfrontとblkback，およびblktapドライバと同様，共有メモリを使って接続される．提案方式では，ドメインUのゲストOSのSCSIサブシステムが前述したような機能を提供するため，新たな機能の実装は必要ない．

blktapドライバ同様，vscsibackドライバは，受け取ったSCSIコマンドを，ユーザ空間のデーモンプロセスに送る．デーモンは，SCSIコマンドを処理し，システムコールを利用して，ディスクI/Oを実行する．このデーモンの処理は，tapdiskデーモンと類似しているため，tapdisk用のスナップショット機能等のコードを容易に再利用できると考えている．

ユーザ空間のデーモンでのSCSIコマンド処理機能（例えば，SCSI READコマンドを適切なシステムコールに変換する）に関しては，我々が，Linuxを使ったSANストレージシステムのためのフレームワークとして開発を進めているtgt[3]を利用することで，実装量を削減する．tgtは，ストレージネットワークを介して，イニシエータから受け取ったSCSIコマンドをユーザ空間のデーモンが処理する仕組みを提供する．

4. まとめ

本稿では，仮想化技術のオープンソースソフトウェアXenのストレージ仮想化方式の課題，および，現在開発中のSCSIプロトコルを使ったストレージ仮想化方式について述べた．

参考文献

[1] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the Art of Virtualization, *The 19th ACM symposium on Operating Systems Principles*, pp. 164–177 (2003).

[2] Warfield, A.: Blktap: Userspace file-based image support. (2006). `http://marc.theaimsgroup.com/?l=xen-devel&m=115073403723020&w=2`.

[3] 藤田智成：Linuxにおけるストレージシステムフレームワークの実現，情報処理学会論文誌：コンピューティングシステム，No. ACS 15（採録決定）(2006).