仮想化システムの外側で動作する シャドウデバイスを考慮したVMマイグレーション

鵜木智矢† 光来健一†

1. はじめに

IaaS 型クラウドではネットワーク経由で仮想マシン（VM）が提供され，ユーザは帯域外リモート管理と呼ばれる管理手法を用いて VM の管理を行う．しかし，クラウドの中には悪意のある管理者が存在する可能性があるため，帯域外リモート管理で用いられる VM の仮想デバイス（仮想キーボードなど）から機密情報が盗まれる危険性がある．そこで，仮想デバイスからの情報漏洩を防ぐために VSBypass[1] が提案されている．VSBypass は，ネストした仮想化を用いて仮想デバイスを仮想化システムの外側で動作させることにより安全な帯域外リモート管理を実現する．この仮想デバイスは**シャドウデバイス**と呼ばれる．しかし，VSBypass では VM をマイグレーションする際にシャドウデバイスのステートが失われるため，マイグレーション後に帯域外リモート管理を継続することができなかった．

本研究では，シャドウデバイスのステートを移送先に転送できるようにすることで，マイグレーション後もシャドウデバイスを用いた帯域外リモート管理を継続することができるシステム *USShadow* を提案する．

2. USShadow

USShadow は図 1 のように，移送元ホストと移送先ホストの仮想化システム内で動作する移送マネージャ間で，仮想化システムの外側にあるシャドウデバイスのステートを転送する．まず，移送元ホストの移送マネージャがシャドウデバイスと通信を行い，シャドウデバイスのステートを取得する．そして，取得したステートを移送先ホストに送信し，シャドウデバイスを停止させる．一方，移送先ホストの移送マネージャは VM マイグレーションの開始時に，新しいシャドウデバイスを起動する．そして，受信したステートをその

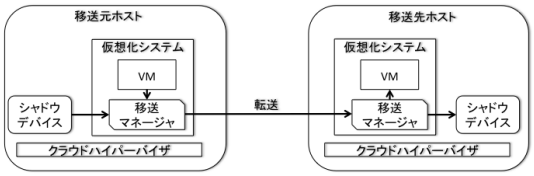

図 1 USShadow における VM マイグレーション

シャドウデバイスへ送り，シャドウデバイスのステートを復元する．このように，シャドウデバイスのステートを保存・復元することで，マイグレーション後に正常にシャドウデバイスを用いることが可能になる．

USShadow では，移送マネージャが仮想化システムの下で動作するクラウドハイパーバイザを直接呼び出すことでシャドウデバイスとの通信を行う．シャドウデバイスにステートの保存・復元コマンドを送信するために，USShadow ではウルトラコール[2] と呼ばれる機構を利用する．ウルトラコールは，CPU の仮想化支援機構を利用し，仮想化システムをバイパスしてクラウドハイパーバイザに直接制御を移す機構である．そのため，仮想化システムの改変が不要であり，既存の仮想化システムを用いることができる．移送マネージャがウルトラコールを用いてクラウドハイパーバイザへコマンドを送信すると，クラウドハイパーバイザは受信したコマンドをイベントに変換してシャドウデバイスに転送する．

コマンドを受信したシャドウデバイスは移送マネージャとの間で共有メモリを確立してステートのやりとりを行う．移送マネージャから SaveState コマンドを受信した場合，シャドウデバイスはこの共有メモリに自身のステートを書き込む．LoadState コマンドを受信した場合には共有メモリからステートを読み込む．共有メモリを用いることにより高速かつ安全な通信が可能になる．

シャドウデバイスのステートの保存・復元の際には，シャドウデバイス自身がステートの暗号化・復号化を行う．シャドウデバイスのステートにはユーザの入力や VM からの出力などの機密情報が含まれることが

† 九州工業大学
Kyushu Institute of Technology

あるためである．シャドウデバイスは仮想化システムの外側で動作するため，シャドウデバイスで暗号化・復号化を行うことで仮想化システム内の管理者への情報漏洩を防ぐことができる．

USShadow では，移送マネージャを改変せずに済ませるために，疑似デバイス経由で透過的にシャドウデバイスのステートの保存・復元を行う．疑似デバイスはシャドウデバイスに対応する仮想デバイスであり，仮想化システム内で動作する．移送マネージャが疑似デバイスのステートを保存しようとした際に，疑似デバイスはシャドウデバイスのステートを取得して，疑似デバイスのステートであるかのように見せかける．同様に，移送マネージャが疑似デバイスのステートを復元しようとした際に，疑似デバイスがシャドウデバイスのステートを復元する．

3. 実　　験

USShadow の動作および性能を確認するための実験を行った．実験には，Xen 4.8 をベースに実装した USShadow を用い，仮想化システムとして Xen 4.4 と KVM 2.4 を動作させた．移送元と移送先のホストには，Intel E3-1226v3 の CPU，8 GB のメモリを搭載したマシンを用い，ギガビットイーサネットで接続した．VM には 2 個の仮想 CPU と 256MB のメモリを割り当てた．

まず，VM マイグレーション後に帯域外リモート管理が継続できるかを確認するために，USShadow と先行研究の VSBypass を用いて VM マイグレーションを行った．実験の結果，USShadow を用いた場合だけ，マイグレーション後も帯域外リモート管理が継続できることを確認した．

次に，USShadow におけるマイグレーション性能を測定し，VSBypass における性能と比較した．マイグレーション時間とダウンタイムを，それぞれ図 2，図 3 に示す．仮想化システムに Xen を用いた場合，USShadow は VSBypass に比べて，マイグレーション時間が 0.9 秒，ダウンタイムが 0.25 秒長くなった．これらは，シャドウデバイスのステートを扱うオーバヘッドと考える．一方，仮想化システムに KVM を用いた場合，Xen を用いた場合と比べて全体的にマイグレーション性能が向上し，USShadow の性能は VSBypass とほぼ変わらなかった．

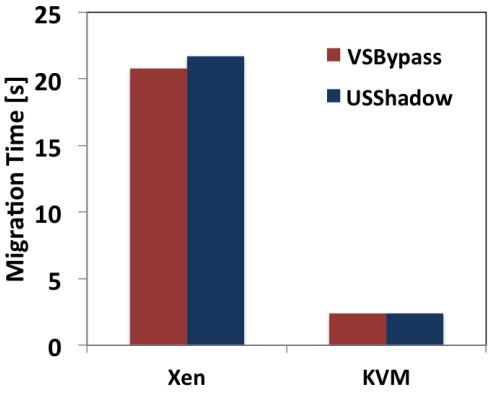

図 2　マイグレーション時間

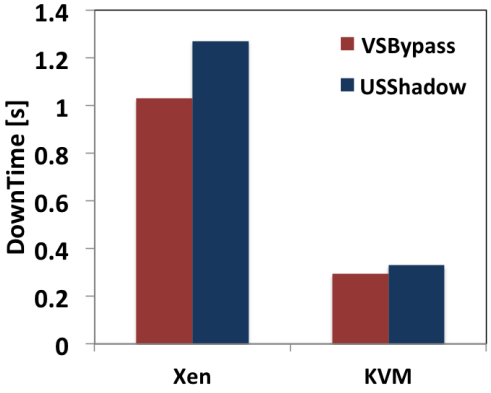

図 3　ダウンタイム

4. ま　と　め

本研究では，マイグレーション後にシャドウデバイスを用いた帯域外リモート管理を継続して行えるシステム USShadow を提案した．現在のところ，シャドウデバイスとして仮想シリアルデバイスにしか対応できていないため，仮想キーボードや仮想ビデオカードにも対応することが今後の課題である．また，シャドウデバイスのステートの暗号化により強力なアルゴリズムを用いることも検討している．

参　考　文　献

1) 二神翔太, 光来健一. VSBypass: ネストした仮想化を用いた VM の安全な帯域外リモート管理. SWoPP2016, 2016.
2) Shohei Miyama and Kenichi Kourai. Secure IDS Offloading with Nested Virtualization and Deep VM Introspection. Proc. ESORICS 2017, 2017.