帯域外リモート管理の継続を可能にするVMマイグレーション

川原　翔[1]　光来 健一[1]

概要：IaaS 型クラウドにおいて，ユーザは提供された仮想マシン（ユーザ VM）をリモート管理する．ユーザ VM を管理する権限を持つ VM（管理 VM）を経由する帯域外リモート管理を行うことで，ユーザ VM の障害発生時でも管理が可能となる．しかし，ユーザ VM を別のホストにマイグレーションすると，管理 VM がユーザ VM にアクセスできなくなり，リモート接続が切断されてしまう．さらに，処理中の入力情報も失われる可能性がある．この問題を解決するために，本稿では帯域外リモート管理の継続が可能な VM マイグレーションを実現するシステム *D-MORE* を提案する．D-MORE は，マイグレーションが可能でユーザ VM にアクセスする特権を持つリモート管理専用の VM である**ドメイン** R を提供し，同期を取りながらドメイン R とユーザ VM を同時にマイグレーションする．D-MORE を Xen に実装し，ユーザ VM をマイグレーションしても帯域外リモート管理が継続できることを確認した．また，マイグレーション時に入力情報が失われず，D-MORE のオーバヘッドが許容範囲内であることを確認した．

1. はじめに

近年，ネットワークを介してユーザにサービスを提供するクラウドコンピューティングの利用が広がっている．そのサービス形態の一つとして，ユーザに仮想マシン（ユーザ VM）を提供する Infrastructure as a Service (IaaS) 型クラウドサービスがある．IaaS 型クラウドを利用することによって，ユーザはハードウェアを用意することなく，必要な時に必要なだけの VM を使用することができる．IaaS 型クラウドのユーザは VNC や SSH などの**リモート管理システム**（*RMS*）を用いて，提供された VM にリモートアクセスすることで，内部のシステムの管理を行う．

IaaS 型クラウドではユーザ VM の障害発生時でも管理を可能とするために**帯域外リモート管理**と呼ばれる管理形態を提供している．この管理形態は従来の帯域内リモート管理と異なり，RMS サーバがユーザ VM 上ではなくユーザ VM を管理する権限を持った VM（管理 VM）上で動作し，仮想キーボードや仮想ビデオカードなどの仮想デバイスを使用して，ユーザ VM に直接アクセスする．この管理形態を用いることで，ユーザ VM のネットワークが VM 内部の設定ミスによって切断されたり，システムクラッシュが起きたりするような障害が発生したとしても，ユーザは VM のリモート管理を継続することができる．

しかし，帯域外リモート管理を行っている場合に，ユーザ VM を別のホストにマイグレーションすると，リモート接続が切断されてしまう．これは，移送元の管理 VM 上の仮想デバイスが削除されると同時にその仮想デバイスを使用している RMS サーバが終了するために発生する．リモート管理を再開するには，ユーザは接続が切断された原因を特定した上で，どのホストの管理 VM に接続し直すかを調べ，再接続を行わなければならない．さらに，リモート管理の入出力情報がマイグレーションによって失われてしまう可能性がある．RMS サーバと仮想デバイスで処理中のデータは RMS サーバと仮想デバイスの終了によって失われる．RMS クライアントと RMS サーバのネットワーク接続も切断されるため，送信中のパケットが失われ再送されない．

この問題を解決するために，本稿では，ユーザ VM のマイグレーション時においても帯域外リモート管理の継続を可能にするシステム *D-MORE* を提案する．D-MORE は，RMS サーバと仮想デバイスを**ドメイン** R と呼ばれるマイグレーションが可能で管理対象のユーザ VM にアクセスする特権を持つリモート管理専用 VM 上で動作させる．帯域外リモート管理の継続を実現するために，D-MORE は RMS クライアント，ドメイン R，管理対象 VM の間の接続をネットワークと仮想マシンモニタ（VMM）レベルで透過的に維持する．さらに，D-MORE はリモート管理の入出力情報が失われることを防ぐことができる．RMS サー

[1] 九州工業大学
Kyushu Institute of Technology

© 1959 Information Processing Society of Japan

バと仮想デバイスで処理中のデータに関してはドメイン R と同時にマイグレーションされる．送信中のパケットはマイグレーション中に失われたとしても TCP によって再送される．

我々は D-MORE を Xen 4.3.2 [2] に実装した．ドメイン R 上で仮想デバイスを動作させるために，ドメイン R はユーザ VM のメモリをマップして共有メモリを確立することができる．さらに，ドメイン R は Xen のイベントチャネルを用いてユーザ VM との間で仮想割り込みチャネルを確立することができる．ドメイン R と管理対象 VM を同時マイグレーションする際には，D-MORE が移送先のホストでメモリのマップ状態とイベントチャネルを復元する．状態の復元を適切なタイミングで行い共有メモリ上のデータ損失を防ぐために，D-MORE はドメイン R と管理対象 VM 間で同期を取りながらマイグレーションを行う．D-MORE を用いた実験を行い，ユーザ VM をマイグレーションしても帯域外リモート管理を継続できることを確認した．さらに，マイグレーション時に入力情報が失われず，同時マイグレーション中のダウンタイムが許容範囲内であることを確認した．

以下， 2 章では，帯域外リモート管理中のマイグレーションによって生じる問題について述べる． 3 章でこの問題を解決する D-MORE について述べ， 4 章でその実装の詳細について述べる． 5 章で D-MORE を用いて行った実験について述べる． 6 章で関連研究に触れ， 7 章で本稿をまとめる．

2. 帯域外リモート管理中のマイグレーション

2.1 帯域外リモート管理

ユーザ VM にネットワーク障害が発生した場合でも管理を行うことができるようにするために，IaaS 型クラウドでは**帯域外リモート管理**を提供することが不可欠になっている．この管理手法では図 **1** に示すように，ユーザ VM ごとに用意された RMS サーバが特権を持つ VM である**管理 VM** 上で動作する．管理 VM は Xen や Hyper-V などのハイパーバイザ型の VMM の多くで提供されており，全てのユーザ VM にアクセスする特権を持つ．さらに，管理 VM は仮想キーボードや仮想ビデオカードといった仮想デバイスをそれぞれのユーザ VM 用に提供する．管理 VM の RMS サーバはユーザ VM が用いる仮想デバイスに直接アクセスすることができる．そのため，ユーザ VM 内の RMS サーバやネットワーク設定に依存しないリモート管理を実現することができる．

これにより，ユーザ VM にネットワーク障害が発生した場合でも，ローカルコンソールからログインしているかのように操作を行うことができる．例えば，ネットワークの設定ミスによってユーザ VM へのネットワーク接続ができなくなったとしても，仮想キーボードを用いて設定ファイルを修正し，ネットワーク接続を復旧させることができる．また，仮想ビデオカードへのアクセスを通して，OS のエラーメッセージを確認することも可能である．

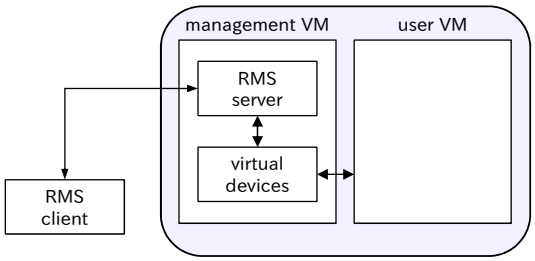


図 1 帯域外リモート管理

2.2 マイグレーション時の問題

IaaS 型クラウドは様々な目的のためにユーザ VM のマイグレーションを行う．しかしながら，帯域外リモート管理中にマイグレーションが行われると，リモート接続が切断されてしまう．ユーザ VM がマイグレーションされる際には，移送元の管理 VM 上の仮想デバイスが削除され，マイグレーションされた VM は移送先のホストの管理 VM 上に作成される新しい仮想デバイスを用いる．同時に移送元のホストの管理 VM 上で動作する RMS サーバは削除された仮想デバイスへのアクセスができなくなるため終了する．結果として，RMS クライアントは RMS サーバから切断されてしまう．リモート管理を再開するには，ユーザはまず，なぜ RMS クライアントが切断されたのかを特定する必要がある．原因としては，ネットワーク障害や管理 VM 内のシステム障害なども考えられる．原因がユーザ VM のマイグレーションだった場合は，ユーザは VM の移送先のホストを特定し，RMS サーバに接続をやり直す必要がある．

さらに，キーボードなどの入力情報がユーザ VM のマイグレーションによって失われる可能性がある．RMS クライアントによって送信された入力情報が RMS サーバによって受け取られていない場合は，そのパケットは失われる．RMS クライアントと RMS サーバ間のネットワーク接続は切断されてしまうため，失われたパケットはネットワークレベルで再送されない．また，RMS サーバによって受け取られた入力情報が仮想デバイスに送られていない場合は，RMS サーバの終了によってデータが失われてしまう．同様に，仮想デバイスによって受け取られたがユーザ VM に送られていない入力情報も，仮想デバイスの削除とともに失われてしまう．キーボード入力情報が失われてしまった場合は，ユーザはもう一度入力をやり直す必要がある．

3. D-MORE

この問題を解決するために，本稿では，帯域外リモート

© 1959 Information Processing Society of Japan

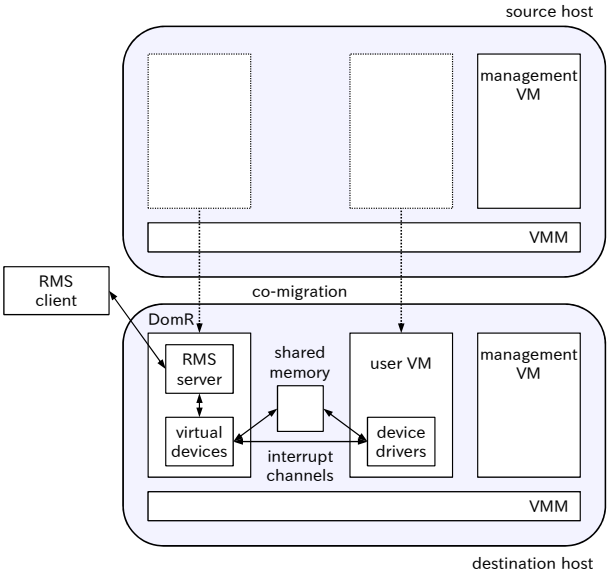


図 2 D-MORE のシステム構成

管理の継続が可能なマイグレーションを実現するシステム *D-MORE* を提案する．D-MORE のシステム構成を**図 2** に示す．D-MORE はマイグレーションが可能でユーザ VM にアクセスする特権を持ったリモート管理専用の VM である**ドメイン R** を提供する．ドメイン R では管理対象 VM 用の RMS サーバおよび仮想デバイスのみを動作させる．RMS クライアントはドメイン R の RMS サーバに接続し，ドメイン R 上で動作する仮想デバイスを用いてユーザ VM にアクセスする．ユーザ VM のマイグレーション時には D-MORE がユーザ VM とドメイン R 間で同期を取りながら同じ移送先ホストに同時マイグレーションする．マイグレーションの間，D-MORE は RMS クライアント，ドメイン R，管理対象 VM の間の全ての接続を透過的に維持する．そのために，D-MORE はドメイン R 上の仮想デバイスと管理対象 VM 間の接続を VMM レベルで維持する．さらに，RMS クライアントとドメイン R 上の RMS サーバ間のネットワーク接続をネットワークレベルで維持する．

ドメイン R は仮想デバイスを動作させるために必要な特権を持った VM である．第一に，ドメイン R は管理対象 VM との間で共有メモリを確立する特権を持っている．共有メモリ上のバッファを用いることで，ドメイン R 上の仮想デバイスは管理対象 VM 上のデバイスドライバと入出力情報の受け渡しを行う．第二に，ドメイン R は管理対象 VM との間で仮想割り込みチャネルを確立する特権を持っている．仮想割り込みチャネルを用いて，ドメイン R 上の仮想デバイスは仮想割り込みを管理対象 VM 上のデバイスドライバに送る．仮想割り込みは新しいデータが共有メモリ上のバッファに書き込まれた際に送られる．

ドメイン R と管理対象 VM の同時マイグレーションが行われる際に，D-MORE は移送先でドメイン R と管理対象 VM 間の接続を再確立する．マイグレーションの間に VM 間の接続はネットワーク接続を除いてすべて失われるため，ドメイン R は一旦，管理対象 VM から切断される．D-MORE はドメイン R と管理対象 VM との間で共有メモリと仮想割り込みチャネルの再確立を行う．そのために，D-MORE はメモリ共有と仮想割り込みチャネルの状態を移送元のホストで保存し，移送先のホストに送信する．移送先ホストでは，D-MORE が保存された状態をドメイン R と管理対象 VM 間で透過的に復元する．そのため，ドメイン R 上の仮想デバイスと管理対象 VM 上のデバイスドライバは共有メモリや仮想割り込みチャネルにアクセス中だったとしても，処理を継続することができる．

D-MORE がドメイン R と管理対象 VM の 2 つのマイグレーション・プロセスの同期を行う目的は 3 つある．第一の目的は，適切なタイミングでドメイン R と管理対象 VM の再接続を行うことである．共有メモリは管理対象 VM のメモリを用いて構築されるため，管理対象 VM のメモリが復元された後で再確立される必要がある．また，整合性を保つために仮想割り込みチャネルは両方の VM の停止中に保存および復元される必要がある．第二の目的は，共有メモリ上の最新のデータが移送先のホストに転送されることを保証することである．ドメイン R がどのタイミングで共有メモリを変更しても，管理対象 VM のマイグレーション・プロセスは最新の情報を転送しなければならない．そのためには，ドメイン R は管理対象 VM がマイグレーションされた後で共有メモリを変更してはならない．第三の目的は，ドメイン R と管理対象 VM の停止をできるだけ遅らせてダウンタイムを削減することである．

D-MORE では，帯域外リモート管理の入力情報が同時マイグレーション中に失われることはない．RMS サーバと仮想デバイスはドメイン R の一部としてマイグレーションされるため，処理中の入力情報は保持される．共有メモリ上のバッファに書き込まれた入力情報は上述の同時マイグレーション中の同期によって保持される．RMS クライアントから RMS サーバに送信中の入力情報に関しては，ドメイン R のマイグレーションによって一時的に失われたとしても TCP によって再送される．RMS サーバがドメイン R 上で動作するため，RMS クライアントとの TCP 接続を維持することが可能である．

4. 実装

我々は D-MORE を Xen 4.3.2 [2] に実装した．Xen において，管理 VM は**ドメイン 0**，ユーザ VM は**ドメイン U** と呼ばれる．ドメイン R はドメイン U を監視することができるドメイン M [10] を拡張して実装した．ドメイン R では仮想デバイスを動作させるために準仮想化 Linux を動作させた．現在の実装では，ドメイン U のゲスト OS として準仮想化 Linux をサポートし，x86-64 アーキテクチャを対象としている．

© 1959 Information Processing Society of Japan

4.1 ドメイン R とドメイン U の通信

ドメイン U が起動すると，D-MORE は新しいドメイン R をドメイン U に対応づける．管理対象のドメイン U との間で共有メモリを確立するために，ドメイン R がドメイン U のメモリページをマップする．以降はドメイン R とドメイン U が同じメモリページにアクセスすることができるようになる．このメモリマップ処理のために，ドメイン R はドメイン U と共有したいメモリページに対応するページフレーム番号を指定してハイパーコールを発行することによって VMM を呼び出す．ドメイン R は対応づけられたドメイン U のメモリページのみをマップすることができる．

加えて，ドメイン R は仮想割り込みチャネルとしてドメイン U との間にイベントチャネルを確立する．イベントチャネルは 2 つの VM 間の論理的な通信路であり，仮想割り込みなどのイベントを送信するために用いられる．イベントチャネルは接続元の VM 番号とポート番号，接続先の VM 番号とポート番号によって構成される．ドメイン U がイベントチャネルを作成してポート番号を割り当てられた後で，ドメイン R はそのポート番号を用いてバインドを行う．この際に，ドメイン U が接続先としてドメイン 0 を指定していた場合でも，ドメイン R はそれを横取りしてイベントチャネルを確立することができる．

4.2 ドメイン R 上の仮想デバイス

帯域外リモート管理に必要な仮想デバイスを提供するために，ドメイン R は Xen 用にカスタマイズされた QEMU [3] を動作させる．VNC によるリモート管理の場合には，仮想キーボード，仮想マウス，仮想フレームバッファが用いられる．SSH によるリモート管理の場合には，仮想コンソールデバイスが用いられる．**図 3** に示すように，準仮想化の仮想デバイスは QEMU におけるバックエンドドライバとして実装されている．ドメイン U 上のフロントエンドドライバは I/O リングとイベントチャネルを用いてこのバックエンドドライバと通信を行う．I/O リングとはデータを受け渡すためのリングバッファであり，共有メモリ上に配置される．フロントエンドドライバおよびバックエンドドライバはデバイスの設定をドメイン 0 上の XenStore 経由で交換する．

フロントエンドドライバとバックエンドドライバの初期化は次のように行われる．ドメイン U 上のフロントエンドドライバがドメイン R 上のバックエンドドライバに接続する際に，I/O リングのためのメモリページを割り当て，XenStore にページフレーム番号を書き込む．次にフロントエンドドライバはイベントチャネルを作成し，割り当てられたポート番号を XenStore に書き込む．一方で，ドメイン R 上のバックエンドドライバは XenStore から I/O リングのページフレーム番号を読み込み，そのページをマップする．その後でドメイン U のイベントチャネルのポート番号を読み込みイベントチャネルを確立する．

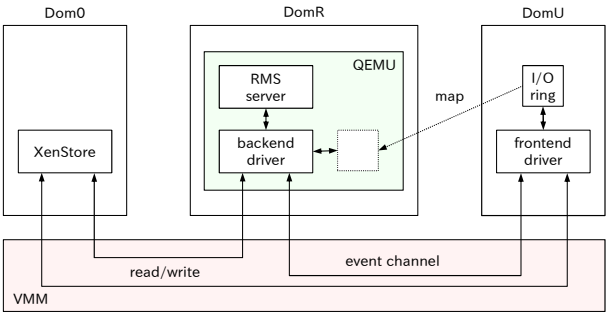


図 3 ドメイン R 上の仮想デバイス

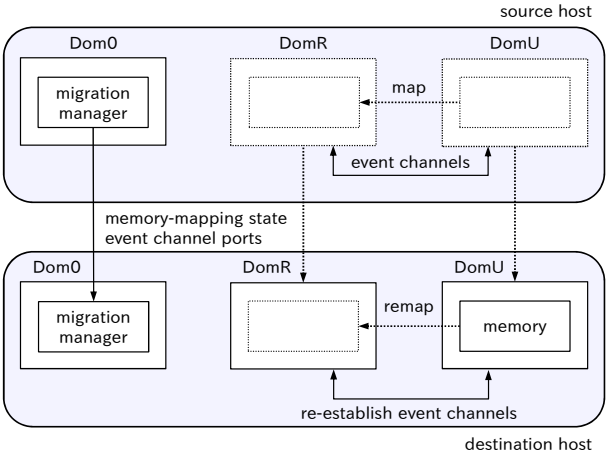


図 4 同時マイグレーション後のメモリマップ状態とイベントチャネルの復元

4.3 ドメイン R とドメイン U の再接続

図 4 に示すように，D-MORE は移送先ホストでドメイン R にマップされたドメイン U のメモリの状態を復元する．そのために，移送元のマイグレーションマネージャはドメイン R のページテーブルを検査し，ドメイン U のメモリページがマップされているページテーブルエントリ（PTE）に監視ビットをセットする．移送先のホストでは，PTE に監視ビットがセットされていた場合，マイグレーションマネージャがドメイン U の対応するメモリページをドメイン R に再マップする．実装の詳細については，先行研究 [10] を参照されたい．

D-MORE は移動先のホストでイベントチャネルの状態の復元も行う．そのために，D-MORE では移送元のマイグレーションマネージャがドメイン R とドメイン U 間で確立されたイベントチャネルのリストを取得し，イベントチャネルに使用されているポートの組を送信する．移送先のホストでは，マイグレーションマネージャがドメイン R とドメイン U 間のイベントチャネルを移動元と同じポートの組を用いて再確立する．

ドメイン R とドメイン U の OS カーネルにおいて再確立されたイベントチャネルを再利用できるようにするため

© 1959 Information Processing Society of Japan

に，仮想割り込みのレジューム処理に修正を行った．準仮想化の Linux カーネルにおいては，割り込み要求（IRQ）とイベントチャネルがマップされている．従来のカーネルにおいては，VM のマイグレーションによってイベントチャネルが閉じられるため，これらのマップ状態は VM のレジュームの際に破棄されていた．D-MORE では，カーネルがハイパーコールを発行することによって再確立されたイベントチャネルの情報を取得し，IRQ とイベントチャネル間のマップ状態を初期化しないようにした．

D-MORE はドメイン R とドメイン U 間で接続を維持するため，ドメイン U 上のフロントエンドドライバのレジューム処理に関しても同様に無効化した．具体的には，I/O リング内にある処理中の入出力データが失われないように I/O リングの再初期化を行わないようにした．I/O リングは同時マイグレーションの直後は整合性が保たれていない可能性があるが，ドメイン R とドメイン U の再開後には整合性が保たれた状態になる．また，D-MORE によって接続が保たれるために，フロントエンドドライバはバックエンドドライバに対して再接続を行わない．

4.4 書き込み可能な共有メモリの転送

ライブマイグレーションは VM を動作させたまま VM のメモリを移送先のホストに転送する．この転送は転送中に書き込みが行われてダーティになったメモリページについて繰り返し行われる．ダーティなメモリページの数が十分に小さくなったら，VM を停止させ，移送先のホストで VM を再開する．ダーティなメモリページを検出するために，Xen では log dirty モードが提供されている．このモードでは VMM がメモリページへの書き込みを検出し，*log dirty* ビットマップに記録する．メモリ転送の各イテレーションの最後で，マイグレーションマネージャは log dirty ビットマップを取得し，ダーティなメモリページのみを次のイテレーションで転送する．

しかし，log dirty モードは共有メモリへの書き込みを正しく検出することはできない．log dirty モードが有効になっているドメイン U に対して，VMM はそのドメイン U によるメモリページへの書き込みのみを検出することができる．つまり，ドメイン R がドメイン U と共有しているメモリページを変更しても，VMM はこれらのページをダーティと認識することができない．そのため，共有メモリがドメイン R によってのみ変更された場合，マイグレーションマネージャは最新の内容を転送することができない．そのため，同時マイグレーション後に I/O リング内の入力データが失われる可能性がある．

このようなデータ損失を防ぐために，D-MORE ではドメイン R によって共有されているドメイン U のメモリページを常にダーティとみなすようにした．これによって，ドメイン R によって変更された共有メモリが転送されることを保証することができる．一方，共有メモリを常にダーティとみなしても，何度も転送されることはない．マイグレーションマネージャは頻繁に更新されていると判断したメモリページの転送を行わないためである．結果として，書き込み可能な共有メモリはライブマイグレーションの最終段階でのみ転送が行われる．

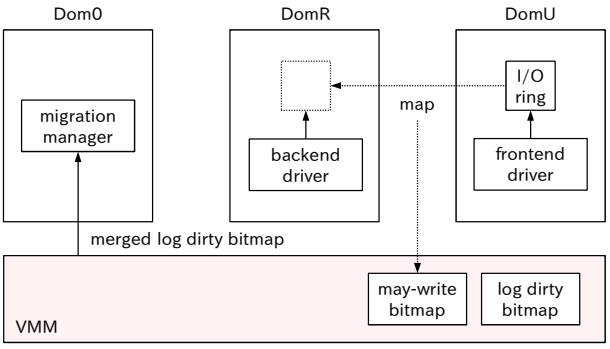


図 5 log dirty モードの拡張

この機能を実現するために，**図 5** に示すように，log dirty モードに拡張を行った．VMM はドメイン U のメモリ共有状態をドメイン U ごとに用意された may-write ビットマップに記録する．ドメイン U のメモリページがドメイン R によって書き込み可能な状態でマップされた場合，VMM は may-write ビットマップの中のページフレーム番号に対応するビットをセットする．マイグレーションマネージャが log dirty ビットマップを取得するためのハイパーコールを発行した時，VMM は log dirty ビットマップに may-write ビットマップをマージして返す．

4.5 同時マイグレーションにおける同期

ドメイン R とドメイン U の同時マイグレーションには**図 6** に示すような 7 つの同期箇所がある．マイグレーションを開始した後，転送された情報を格納するために移動先のホストに新しい空の VM が作成される．ドメイン R が作成された後の S_1 において，ドメイン U の作成を待ち，作成されたドメイン U をドメイン R の管理対象として登録する．

次に，S_2 において同期をとってライブマイグレーションの最終段階に入る．ダウンタイムを削減するために，それぞれのマイグレーションマネージャはもう一方の状態を確認しながら，同時に最終段階に入れるようになるまでメモリ転送を繰り返す．次の S_3 では，ドメイン U のマイグレーションマネージャがドメイン R の停止を待つ．これによってドメイン U が停止した後にドメイン R が共有メモリを変更しないこと，および，共有メモリ上のデータが確実に転送されることを保証する．

ドメイン R のマイグレーションマネージャはドメイン U が停止するのを S_4 で待ち，イベントチャネルで使われているポートの組を保存する．他方で，ドメイン U のマイグ

© 1959 Information Processing Society of Japan

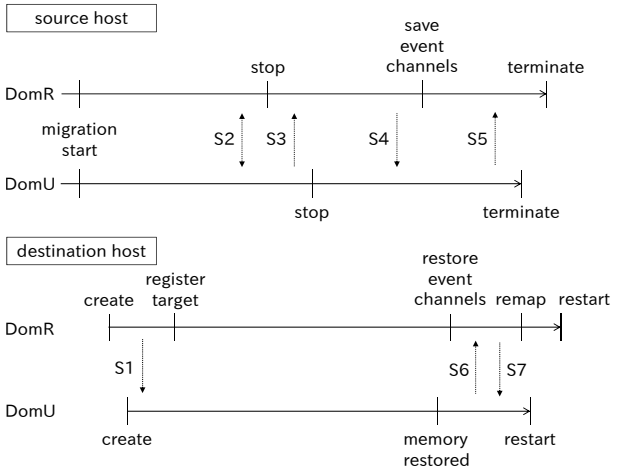


図 6 ドメイン R とドメイン U の同時マイグレーション時の同期

レーションマネージャはイベントチャネルの保存が終わるまで S_5 で待つ．また，移送先のホストにおいて，ドメイン U を再開する前に，イベントチャネルの復元が 完了するのを S_6 で待つ．これらの 3 つの同期によって，VM が停止した状態でイベントチャネルの状態が保存，復元されることを保証する．

S_7 において，ドメイン R のマイグレーションマネージャはドメイン U の全てのメモリが復元されるのを待つ．その後で，ドメイン U に割り当てられたメモリページの一覧を取得し，ドメイン U のメモリをドメイン R に再マップする．

5. 実験

D-MORE を用いてユーザ VM をマイグレーションしても帯域外リモート管理を継続できることを確認し，リモート管理と同時マイグレーションの性能を調べるための実験を行った．サーバ用に，Intel Xeon E3-1270 3.40GHz の CPU，8GB のメモリを搭載したマシンを 2 台用意した．仮想化ソフトウェアとして Xen 4.3.2 を用い，ドメイン 0，ドメイン U，ドメイン R で Linux 3.7.10 を動作させた．デフォルトでドメイン R には 1 個の仮想 CPU と 128MB のメモリ，ドメイン U には 1 個の仮想 CPU と 2GB のメモリ，ドメイン 0 には 8 個の仮想 CPU と残りのメモリを割り当てた．VNC のクライアント用に Intel Xeon E5-1620 3.60 GHz の CPU，8GB のメモリを搭載したマシンを 1 台用意し，TightVNC Java Viewer 2.0.95 [16] を Windows 7 上で動作させた．SSH のクライアント用に OpenSSH 6.0 を移送元のマシン上で動作させた．これらのマシンはギガビットイーサネット・スイッチで接続した．

5.1 同時マイグレーション中のデータ損失

帯域外リモート管理中にマイグレーションを行った時にデータ損失を防止することができるかを確認する実験を行った．VNC クライアントから VNC サーバに 50 ミリ秒ごとにキーを送り続け，仮想キーボードのバックエンドドライバが受け取ったデータを監視した．この実験を 10 回行い，失われたキー入力の数を集計した．従来の Xen ではマイグレーションによってバックエンドドライバが削除されることにより，平均 1.4 個のキー入力が失われた．D-MORE においてはバックエンドドライバもマイグレーションされるため，キー入力は全く失われなかった．マイグレーションに起因する TCP 再送の回数は平均で 8.5 回であった．

次に，共有メモリに書き込まれたデータの損失を防止することができるかを確認する実験を行った．同時マイグレーションを 10 回行い，log dirty モードへの拡張を無効にした場合にデータが失われるかどうかの調査を行った．標準ではデータ損失を確認することができなかったが，ドメイン U のイベントチャネルの読み込みに 200ms の遅延を加えたところ，キー入力が失われることが確認できた．この結果より，スケジューリングによっては I/O リング内のデータが失われる可能性があると考えられる．

5.2 ドメイン R のオーバヘッド

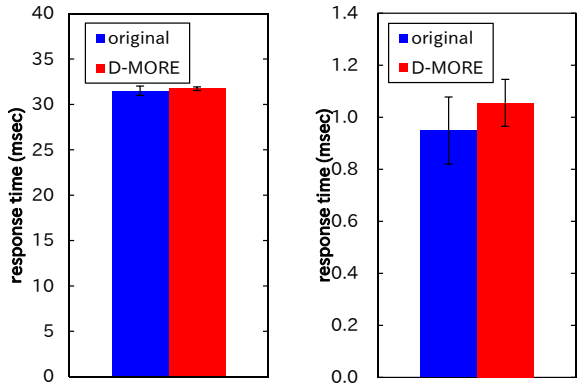
ドメイン R を経由した帯域外リモート管理のオーバヘッドを調べるために，まず VNC および SSH における入力の応答時間を測定した．応答時間はこれらのクライアントがサーバに入力情報を送信し，リモートエコーを受け取るまでの時間とした．比較のためにドメイン 0 で VNC サーバおよび SSH サーバを動作させる従来システムにおける応答時間も測定した．VNC と SSH における応答時間を**図 7a**，**図 7b** にそれぞれ示す． SSH の場合には D-MORE のほうが若干，応答時間が増加しているが，0.1ms 程度であり，ほとんど影響はないと考えられる．

次に，VNC および SSH における出力のスループットを測定した．VNC においては，ドメイン U で全画面 (800×600) を頻繁に書き換えるスクリーンセーバーを動作させ，VNC クライアントにおける画面更新のフレームレートを測定した．SSH においては，cat コマンドで 1000 万文字のテキストファイルを表示させ，SSH クライアントにおける 1 秒当たりの表示文字数を測定した．いずれも D-MORE および従来システムについて測定した．VNC と SSH におけるスループットを**図 8a**，**図 8b** にそれぞれ示す．VNC では同程度のスループットであったが，SSH では D-MORE のほうが若干スループットが低くなることが分かった．

5.3 同時マイグレーション時間

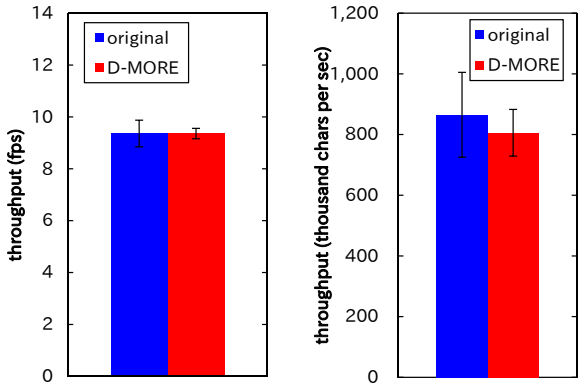
ドメイン R とドメイン U の同時マイグレーションに要する時間を測定した．この実験においてはドメイン U のメモリサイズを 256MB から 2GB まで変化させた．帯域外リモート管理の影響を調べるため，VNC クライアントおよび SSH クライアントからサーバに 50ms ごとに入力情

© 1959 Information Processing Society of Japan



(a) VNC　　(b) SSH

図 7　入力の応答時間



(a) VNC　　(b) SSH

図 8　出力のスループット

報を送った場合と送らなかった場合について測定した．比較として，独立したドメイン U を同期せずに同時にマイグレーションした場合についても測定した．同時マイグレーション時間はマイグレーションを開始してから両方の VM のマイグレーションが完了するまでの時間とした．

VNC と SSH でのリモート管理中に同時マイグレーションを 10 回ずつ行った時の平均をそれぞれ**図 9**，**図 10** に示す．同時マイグレーション時間はドメイン U のメモリサイズに比例して増加した．2 つのドメイン U の独立したマイグレーションと比較して，D-MORE の同時マイグレーション時間は VNC の場合で 1.6 秒，SSH の場合で 0.4 秒しか増加しなかった．一方，同時マイグレーション中にリモート接続を用いて入力情報を送信した場合，VNC の場合で 4.2 秒，SSH の場合で最大 3.2 秒マイグレーション時間が増加した．これはドメイン R とドメイン U において多くのメモリがダーティになったためと考えられる．

5.4　ダウンタイム

VNC を用いた場合に，同時マイグレーション中のドメイン R とドメイン U のダウンタイムを測定する実験を行った．VM が移送元ホストと移送先ホストのどちらでも動作していない時間をダウンタイムとした．この実験を 10 回

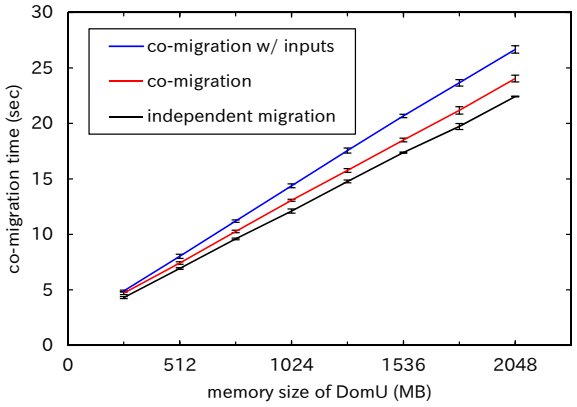


図 9　VNC によるリモート管理中の同時マイグレーション時間

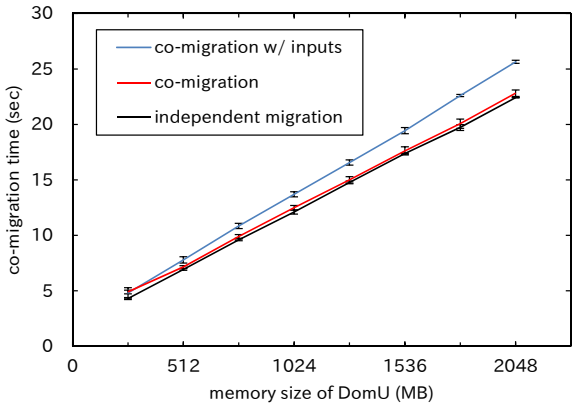


図 10　SSH によるリモート管理中の同時マイグレーション時間

行った時の平均のダウンタイムを**図 11** に示す．これらのダウンタイムは分散が非常に大きいため，見やすさのためにエラーバーを省いた．

ドメイン U については，独立したマイグレーションの場合より D-MORE の場合の方がダウンタイムが短かった．一方，ドメイン R についてはドメイン U よりも前に停止され，後で再開されたため，ダウンタイムはドメイン U よりも大きくなった．リモート接続を用いて入力情報を送信しながら同時マイグレーションを行っても，その影響は小さかった．

次に，ユーザが RMS クライアントを使う時に感じるダウンタイムを測定した．VNC と SSH のクライアントで 50ms ごとにキーを送り，同時マイグレーションの最終段階における最長の応答時間をユーザが感じるダウンタイムとした．このダウンタイムを 10 回測定した時の平均を**図 12** に示す．VNC において，このダウンタイムはドメイン U のメモリサイズが大きくなるほど小さくなる傾向にあり，その最大値は 827ms であった．一方，SSH においてはドメイン U のメモリサイズによらずほぼ一定であり，最大値は 613ms であった．

5.5　同時マイグレーション中の性能低下

同時マイグレーションは帯域外リモート管理の性能に影

© 1959 Information Processing Society of Japan

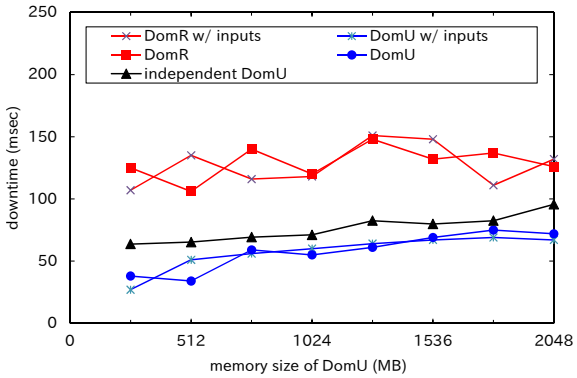


図 11　VM のダウンタイム

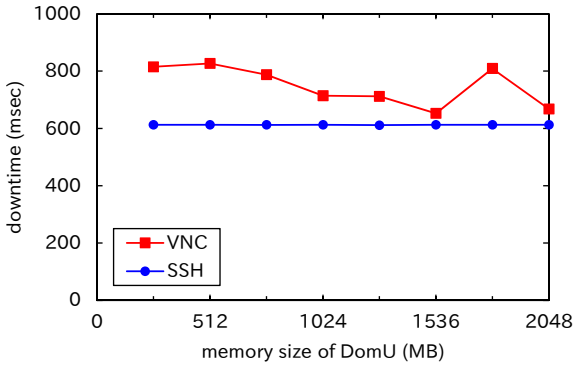


図 12　RMS クライアントにおけるダウンタイム

響を与えるため，同時マイグレーション中の応答時間の変化を測定した．この実験においては，ドメイン U に 2GB のメモリを割り当て，50ms ごとにキー入力情報を送信し続けた．VNC と SSH における応答時間を**図 13**，**図 14** にそれぞれ示す．同時マイグレーションを開始すると，応答時間が VNC の場合で平均 5.4ms，SSH の場合で平均 2.0ms 増加した．この性能低下は 30 秒間続いた．

次に，同時マイグレーション中の VNC のスループットの変化を測定した．この実験では 5.2 節のスクリーンセーバを用いた．**図 15** に示すように，同時マイグレーションを開始するとフレームレートは平均 0.4fps 低下し，フレームレートの低下は 30 秒間続いた．

6. 関連研究

VM のマイグレーション時に帯域外リモート管理を継続する方法として，D-MORE とは異なる 2 つの手法がある．一つは VNC プロキシを用いる方法である．この手法では，ユーザ VM を管理するために，VNC クライアントは VNC プロキシを経由して管理 VM 内の VNC サーバにアクセスする．ユーザ VM がマイグレーションされた際には，VNC プロキシが接続を移送先の VNC サーバに透過的に切り替えることができる．しかし，VNC サーバと仮想デバイスで処理中の全ての情報は失われてしまう．

もう一つはマイグレーションをプロトコルレベルでサポートしている SPICE [14] を用いる方法である．VM がマイグレーションされると，管理 VM 上の SPICE サーバは SPICE クライアントに移送先のホストを通知し，SPICE クライアントが移送先の SPICE サーバに接続を切り替える．この際に，SPICE のシームレスマイグレーションを用いればデータは失わない．SPICE の欠点としては，VM の管理が特定の RMS に依存するということである．D-MORE は VNC や SSH などの様々な RMS を用いてリモート管理を行うことができる．

ドメイン R に似た特徴を持つ VM も提案されている．Xen のスタブドメイン [13], [15] や Xoar [6] の Qemu VM

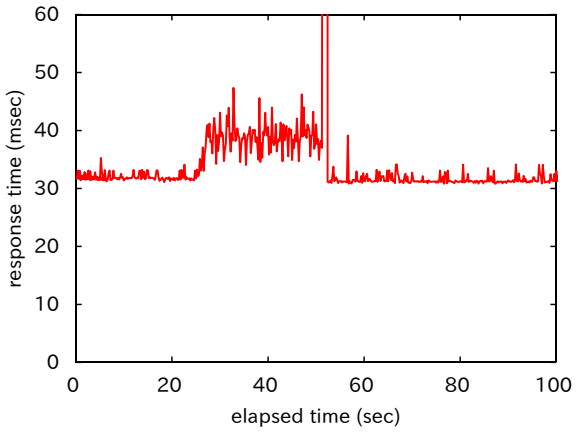


図 13　同時マイグレーション中の VNC の応答時間の変化

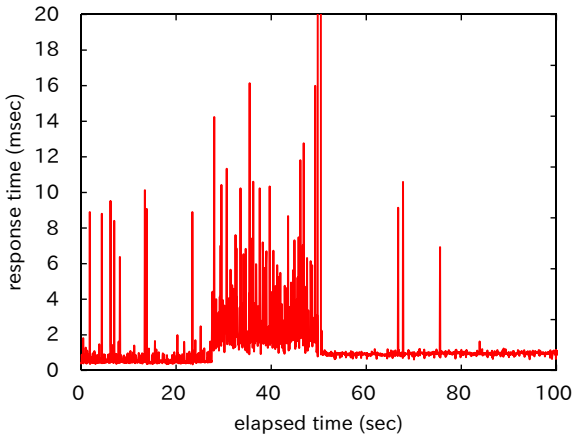


図 14　同時マイグレーション中の SSH の応答時間の変化

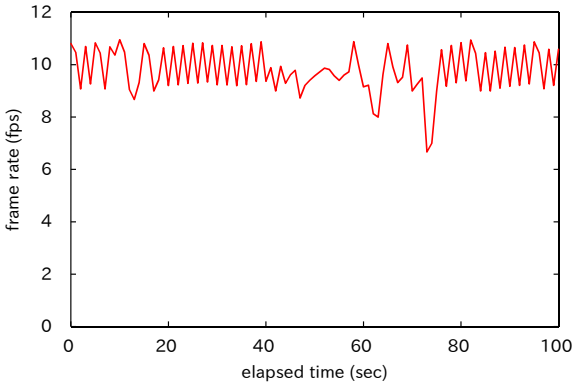


図 15　同時マイグレーション中のフレームレートの変化

© 1959 Information Processing Society of Japan

はドメイン U に仮想デバイスを提供するために QEMU を動作させることができる．これらの VM は完全仮想化のドメイン U のみをサポートしている．Xen のドライバドメイン [8] は準仮想化カーネル内でバックエンドドライバを動作させることができる．VMCoupler [10] のドメイン M はユーザ VM にアクセスする特権を持ち，マイグレーションが可能な VM である．しかし，ドメイン R と違い，イベントチャネルを横取りすることはできない．SSC [4] のサービスドメインは，VM のメモリを監視する特権を持つ．ドメイン B [12] はドメイン U を安全に起動するためにドメイン U のメモリにカーネルイメージを読み込む．

D-MORE と同様に，VMCoupler [10] はドメイン M と監視対象 VM を同期しながら同時マイグレーションすることができる．VMCoupler はセキュリティを目的として二つのマイグレーションプロセスを同期するのに対し，D-MORE は帯域外リモート管理の継続を目的としている．そのため，同時マイグレーションの同期は VMCoupler と D-MORE で大きく異なる．VMCoupler は VM の状態だけを用いて 4 箇所のみで同期を行う．一方，D-MORE はマイグレーションマネージャにおける様々な状態を用いて 7 箇所で同期を行う．

複数の VM を並列にマイグレーションするために，ライブギャングマイグレーション [7] が提案されている．この手法はマイグレーションのオーバヘッドを削減するために VM 間にまたがって同一の内容のメモリページを一回だけ転送する．さらに，ほとんど同一のメモリページに対して差分を利用した圧縮を行う．D-MORE と異なり，ライブギャングマイグレーションは VM 間でマイグレーションプロセスの同期を行わない．

7. まとめ

本稿では，ユーザ VM のマイグレーション時においても帯域外リモート管理の継続を可能とするシステム D-MORE を提案した．D-MORE はドメイン R で RMS サーバと仮想デバイスを動作させる．そして，ドメイン R と管理対象 VM の同期を取りながら同時マイグレーションし，RMS クライアント，ドメイン R，管理対象 VM の間の接続を VMM レベルおよびネットワークレベルで維持する．D-MORE を Xen に実装し，ユーザ VM のリモート管理が切断されないことを確認した．さらに，処理中のデータが失われることがなく，D-MORE のオーバヘッドが許容範囲内であることを確認した．

今後の課題は D-MORE において完全仮想化のゲスト OS に対応することである．また，Xen の Mini OS などを用いることでドメイン R のリソース消費量を削減することも今後の課題である．

参考文献

[1] Apache Software Foundation. Apache CloudStack: Open Source Cloud Computing. `http://cloudstack.apache.org/`.
[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *Proc. Symp. Operating Systems Principles*, pp. 164–177, 2003.
[3] F. Bellard. QEMU. `http://qemu.org/`.
[4] S. Butt, H. A. Lagar-Cavilla, A. Srivastava, and V. Ganapathy. Self-service Cloud Computing. In *Proc. Conf. Computer and Communications Security*, pp. 253–264, 2012.
[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. In *Proc. Symp. Networked Systems Design and Implementation*, pp. 273–286, 2005.
[6] P. Colp, M. Nanavati, J. Zhu, W. Aiello, G. Coker, T. Deegan, P. Loscocco, and A. Warfield. Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor. In *Proc. Symp. Operating Systems Principles*, pp. 189–202, 2011.
[7] U. Deshpande, X. Wang, and K. Gopalan. Live Gang Migration of Virtual Machines. In *Proc. Intl. Symp. High Performance Distributed Computing*, pp. 135–146, 2011.
[8] K. Fraser, S. Hand, R. Neugebauer, I. Pratt, A. Warfield, and M. Williamson. Safe Hardware Access with the Xen Virtual Machine Monitor. In *Proc. Workshop on Operating System and Architectural Support for the on demand IT InfraStructure*, 2004.
[9] A. Kivity and M. Tosatti. Kernel Based Virtual Machine. `http://www.linux-kvm.org/`, 2007.
[10] K. Kourai and H. Utsunomiya. Synchronized Comigration of Virtual Machines for IDS Offloading in Clouds. In *Proc. Intl. Conf. Cloud Computing Technology and Science*, pp. 120–129, 2013.
[11] D. S. Milojíčić, F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou. Process Migration. *ACM Comput. Surv.*, Vol. 32, No. 3, pp. 241–299, 2000.
[12] D. G. Murray, G. Milos, and S. Hand. Improving Xen Security through Disaggregation. In *Proc. Intl. Conf. Virtual Execution Environments*, pp. 151–160, 2008.
[13] J. Nakajima and D. Stekloff. Improving HVM Domain Isolation and Performance. In *Xen Summit September 2006*, 2006.
[14] Red Hat. Spice. `http://www.spice-space.org/`.
[15] S. Thibault. Stub Domains. In *Xen Summit Boston 2008*, 2008.
[16] TightVNC Group. TightVNC. `http://www.tightvnc.com/`.
[17] VMware Inc. VMware vSphere Hypervisor. `http://www.vmware.com/`.

© 1959 Information Processing Society of Japan