IPmigrate：複数ホストに分割されたVMのマイグレーション手法

柏木　崇広[1,a)]　末竹　将人[1]　光来　健一[1]

概要：近年，ユーザに仮想マシン (VM) を提供する IaaS 型クラウドが普及しており，大容量メモリを持つ VM も提供されるようになってきた．大容量メモリを持つ VM のマイグレーションを容易にするために，複数のホストに VM のメモリを分割して転送する分割マイグレーションが提案されている．しかし，分割マイグレーション後は VM が他のホストのメモリを必要とするたびにリモートページングが発生し，VM の性能が低下する．また，複数ホストの内，一部のホストのみをメンテナンスする場合でも，VM 全体を停止させる必要がある．本稿では，複数のホストに分割された VM のマイグレーションを可能にするシステム *IPmigrate* を提案する． IPmigrate では，分割された VM を再び一つのホストで動作させる**統合マイグレーション**と，VM の一部だけのマイグレーションを行う**部分マイグレーション**を提供する．これらのマイグレーション中にリモートページングが発生した場合でも，IPmigrate は当該ページの再送や無効化を行うことにより過不足なくメモリを転送する．我々は統合マイグレーションとメインホスト間での部分マイグレーションを KVM に実装し，マイグレーション性能およびマイグレーション後の VM の性能を測定した．

1. はじめに

近年，IaaS 型クラウドでは，大容量メモリを持つ VM も提供されるようになってきている．例えば，Amazon EC2 では 2TB のメモリを持つ VM が提供されている．このような大容量メモリを持つ VM を用いることでより高速にビッグデータの解析 [1][2] をすることが可能となる．VM を用いる利点として，ホストをメンテナンスする際に VM のマイグレーションによるサービスの継続が可能になることが挙げられる．しかし，大容量メモリを持つ VM をマイグレーションする際には，移送先ホストとして十分な空きメモリを持つホストが存在しない場合がある．このような場合には，移送先ホストの仮想メモリを活用することによりマイグレーションを行うことができるが，マイグレーション中やマイグレーション後のページングにより，VM の性能が大幅に低下する．

そこで，大容量メモリを持つ VM を複数ホストに分割して転送する分割マイグレーション [3] が提案されている．分割マイグレーションでは，VM のメモリを 1 台のメインホストと複数のサブホストに転送する．メインホストには，アクセスされることが予測されるメモリおよび，CPU やデバイスの状態等の VM の核となる情報を転送し，サブホストにはメインホストに入りきらないメモリを転送する．分割マイグレーション後にはメインホスト上で VM を動作させ，VM がサブホスト上に存在するメモリを必要とした時にはメインホストとサブホスト間でリモートページングを行う．しかし，分割したまま VM を実行し続けるとネットワーク転送を伴うリモートページングのオーバヘッドのために 1 台のホストで動作する VM と比べて性能が低下してしまう．また，VM が動作している複数のホストの内，一部のホストをメンテナンスする際でも VM 全体を停止させなければならない．

そこで本稿では，これらの問題点を解決するために複数ホストにまたがって動作する VM のマイグレーションを可能にするシステム *IPmigrate* を提案する．IPmigrate は，複数ホストに分割された VM を一つのホストに統合する**統合マイグレーション**を可能にする．統合マイグレーションではメインホストとサブホストから VM のメモリを並列に転送し，リモートページングを行わずに一つの VM として動作できるようにする．また，IPmigrate は分割された VM の一部を別のホスト上に移動させる**部分マイグレーション**も可能にする．メインホスト間やサブホスト間などでのマイグレーションを行うことができ，一部のホストのメンテナンスに柔軟に対応することができる．これらのマ

[1] 九州工業大学
Kyushu Institute of Technology
[a)] kashiwagi@ksl.ci.kyutech.ac.jp

イグレーション中にリモートページングが発生することがあるが，その場合でもメモリを過不足なく転送し，必要に応じて転送済みメモリの無効化を行うことで整合性を保つ．

我々は，IPmigrate を KVM に実装し，統合マイグレーションとメインホスト間での部分マイグレーションを実現した．統合マイグレーションでは，移送元メインホストと移送元サブホストから並列にメモリデータを転送できるようにするために，QEMU-KVM のマイグレーション機構および，サブホストで動作するメモリサーバへの拡張を行った．部分マイグレーションでは，移送元メインホストに存在する VM のメモリだけを転送できるようにした．マイグレーション中にサブホストからメインホストへのページインが発生した場合には，そのメモリページを再送することですべてのメモリが転送されることを保証する．実験の結果，統合マイグレーションはメモリの並列転送により従来のマイグレーションより高速化できることが分かった．

以下，2 章で分割マイグレーションとその問題点について述べ，3 章で複数ホストにまたがって動作する VM のマイグレーションを可能にするシステム *IPmigrate* を提案する．4 章で IPmigrate の実装について説明し，5 章で IPmigrate を用いて行った実験について述べる．6 章で関連研究について触れ，7 章で本稿をまとめる．

2. 分割マイグレーション

2.1 大容量メモリを持つ VM のマイグレーション

VM マイグレーションは，VM を停止させることなく別のホストに移動させる技術である．マイグレーションを用いることで，サービスを止めることなくホストのメンテナンスを行うことができる．マイグレーションを行う際にはまず，移送先ホストに VM を作成し，その後，移送元ホストの VM のメモリをネットワーク経由で移送先ホストへ転送していく．転送中に変更された VM のメモリは移送先ホストに再送され，再送されるメモリが十分小さくなったら移送元ホストの VM を停止させる．そして，VM の変更されたメモリの残りや CPU，デバイスの状態を転送し，移送先ホストで VM の実行を再開する．

近年，大容量メモリを持つ VM が利用されるようになってきており，Amazon EC2 では 2TB のメモリを持つ VM が提供されている．マイグレーションを行う際には移送先ホストに VM のメモリよりも大きな空きメモリが必要となるが，大容量メモリを持つ VM の場合には適切な移送先ホストを見つけるのはより困難になる，これは，常に十分な空きメモリを持ったホストを確保し続けることはコストの面から困難なためである．移送先として適切なホストが存在しない場合，VM のマイグレーションを行うことができないため，ホストのメンテナンスの間，ユーザは VM のサービスを利用することができなくなってしまう．

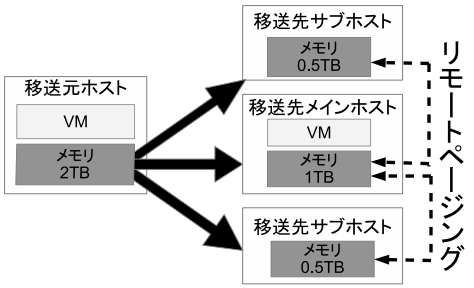

図 1 分割マイグレーション

2.2 S-memV

そこで，**図 1** のように大容量メモリを持つ VM を複数のホストに分割してマイグレーションを行うシステム S-memV[3] が提案されている．S-memV では，マイグレーション後に VM を動作させるホストをメインホスト，メインホスト以外をサブホストと呼ぶ．S-memV は，CPU やデバイスの状態などの VM の核となる情報および，アクセスされることが予測されるメモリをメインホストに転送し，メインホストに入りきらないメモリはサブホストに転送する．移送元ホストからそれぞれのホストにメモリを直接転送することにより，マイグレーション中にはリモートページングが発生しない．S-memV では，VM のメモリの参照履歴を管理することにより，LRU に基づいてメモリを分割する．

マイグレーション後は，メインホスト上の VM はサブホストとの間でネットワーク越しにリモートページングを行いながら動作する．VM がサブホストに存在するメモリを必要とした場合には，当該メモリをサブホストからメインホストに転送（ページイン）する．その代わりに，メモリの参照履歴に基づいて今後 VM がアクセスしないことが予測されるメインメモリ上のメモリをサブホストへ転送（ページアウト）する．分割マイグレーションの際にアクセスすることが予測されるメモリはあらかじめメインホストに転送済みであるため，マイグレーション直後のリモートページングの頻度は抑えられることが期待できる．

しかし，一つの VM を複数ホストに分割するといくつかの問題が生じる．第一に，リモートページングによる VM 性能の低下が挙げられる．VM がメインホスト上に存在しないメモリを必要とした場合にはリモートページングが行われ，メインホストとサブホスト間でページのネットワーク転送が起こる．高速なネットワークであってもメモリよりははるかに遅いため，VM の性能に大きな影響が及ぶ．そのため，分割マイグレーションを行う前に VM が動作していたホストのメンテナンスが終了した後や，十分な空きメモリを持つホストが確保できた場合には，再び一つのホストで VM を動作させることが望ましい．

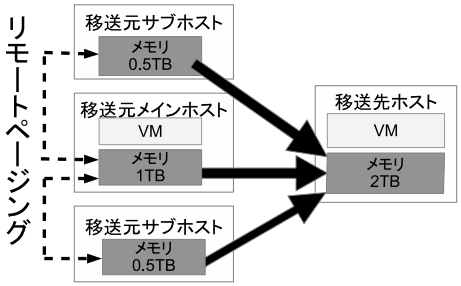

図 2 統合マイグレーション

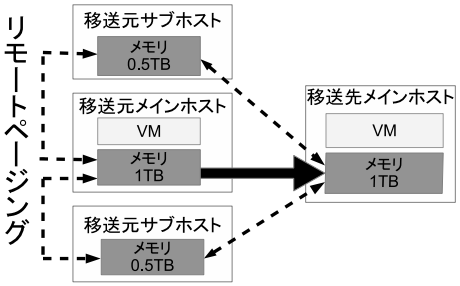

図 3 部分マイグレーションの一例

第二に，分割マイグレーション後に VM が動作している複数ホストの内，一部のホストをメンテナンスしたい場合でも VM 全体を停止させる必要がある．これにより，メンテナンス時でもサービスの提供を継続できるという VM の利点が失われてしまう．このような場合には，メンテナンス対象のホストだけを別のホストに置き換えることで VM が実行し続けられるようにするべきである．

3. IPmigrate

本稿では，複数ホストにまたがって動作する VM に対するマイグレーションを可能にするシステム IPmigrate を提案する．

3.1 統合マイグレーション

統合マイグレーションは，**図 2** のように複数ホストにまたがって動作する VM を一つのホストに統合するマイグレーションである．十分な空きメモリを持つホストが存在する時には，分割された VM をそのホストにマイグレーションすることで，リモートページングによる VM の性能低下を防ぐことができる．統合マイグレーションでは，移送元メインホストは VM の核となる情報とメインホストに存在する VM のメモリを移送先ホストに転送する．同時に，移送元サブホストに存在する VM のメモリを移送先ホストへ転送する．移送元メインホストと移送元サブホストから並列にデータを転送することにより，マイグレーションの高速化を図る．移送先ホストでは VM の全メモリを受信するのを待って，VM の実行を再開する．

統合マイグレーション中にリモートページングが発生した場合でも，IPmigrate はメモリの転送を過不足なく行う．マイグレーション中にサブホストからメインホストへのページインが発生した場合には，必要に応じて移送元メインホストが当該ページを移送先ホストに転送する．転送する必要があるのは，当該ページがいずれのホストからも未転送である場合，および，メインホストで更新された後に転送されていない場合である．一方，マイグレーション中にメインホストからサブホストへのページアウトが発生した場合には，必要に応じて移送元サブホストが当該ページを移送先ホストに転送する．転送する条件はページインされたページの場合と同一である．IPmigrate はメモリの過不足ない転送に必要な情報をメインホストとサブホストにまたがって管理する．

3.2 部分マイグレーション

部分マイグレーションは，VM が動作している複数ホストの内，指定したホスト上にある VM の一部だけを別のホストに移動するマイグレーションである．部分マイグレーションを用いることで，VM を停止させることなく，一部のホストのメンテナンスを行うことができる．例えば，メインホスト間での部分マイグレーションの場合，**図 3** のように移送元メインホストに存在する VM のメモリおよび，VM の核となる情報のみを移送先メインホストへ転送する．この時，サブホストに存在するメモリの転送は行わず，そのメモリに関する情報のみを移送先ホストへ転送する．部分マイグレーションが完了すると，移送先メインホストはマイグレーションされていない元のサブホストとの間でリモートページングを行いながら VM の実行を行う．

一方，サブホスト間での部分マイグレーションの場合，移送元サブホストに存在する VM のメモリだけを移送先サブホストへ転送し，メインホスト上の VM の状態については転送を行わない．マイグレーション後には，元のメインホストが移送先サブホストとリモートページングを行いながら VM を実行する．

部分マイグレーション中に，リモートページングが発生した場合には，IPmigrate はメモリを過不足なく転送するだけでなく，メモリの整合性を保つための処理も行う．メインホスト間での部分マイグレーション中にメインホストへのページインが発生した場合には，必要に応じて移送元メインホストから移送先メインホストへ当該ページを転送する．転送する必要があるのは，当該ページが未転送である場合，および，更新された後に転送されていない場合である．一方，このマイグレーション中にサブホストへのページアウトが発生した場合には，そのサブホストはマイ

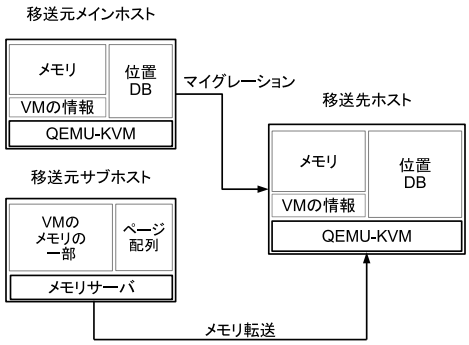

図 4　IPmigrate のシステム構成

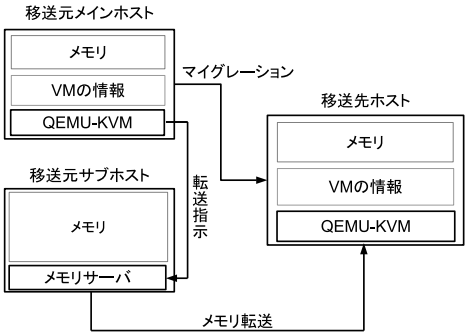

図 5　統合マイグレーションの流れ

グレーション後も動作し続けるため，当該ページの転送は行わない．ただし，そのページを移送先メインホストへ転送済みであれば，同じページがサブホストと移送先メインホストの両方に存在するのを防ぐために，移送先メインホストで当該ページの無効化を行う．

サブホスト間での部分マイグレーション中にメインホストからのページアウトが発生した場合には，必要に応じて移送元サブホストが当該ページを移送先サブホストに転送する．転送する条件は当該ページが未転送の場合，および，変更後に転送されていない場合である．一方，メインホストへのページインが発生した場合には，当該ページがすでに移送先サブホストに転送されていた場合には無効化を行う．

4. 実装

IPmigrate を QEMU-KVM 2.4.1 および Linux 4.3 に実装した．現在のところ，統合マイグレーションとメインホスト間での部分マイグレーションに対応している．

4.1 システム構成

IPmigrate のシステム構成は図 **4** のようになる．移送元メインホストと移送先 (メイン) ホストでは，IPmigrate を実装した QEMU-KVM を動作させ，移送元サブホストでは VM のメモリの一部を管理するメモリサーバを動作させる．移送元メインホストはメモリページがどのホストに存在するかという情報が登録された位置データベースを管理する．位置データベースは分割マイグレーションの際に作成され，サブホストとの間でリモートページングを行うたびに更新される．サブホストからのページインが行われると，当該ページがメインホストに存在するという情報が位置データベースに登録される．逆に，サブホストへのページアウトを行うと，当該ページがページアウト先のサブホストに存在するという情報が登録される．

移送元サブホストは VM のメモリを管理するためにページ配列を管理する．ページ配列にはメモリページがページフレーム番号とメモリデータの組で登録される．ページ配列も分割マイグレーションの際に作成され，メインホストとの間でリモートページングが行われるたびに更新される．メインホストへのページインを行うと，当該ページの情報はページ配列から削除される．逆に，メインホストからのページアウトを行うと，当該ページの情報がページ配列に追加される．

4.2 統合マイグレーションの流れ

統合マイグレーションは図 **5** のように行われる．複数ホストにまたがった VM を一つのホストにマイグレーションできるようにするために，QEMU-KVM のマイグレーション機構の拡張を行った．移送元メインホストの QEMU-KVM は，位置データベースに基づいてメインホスト上に存在する VM のメモリページだけを移送先ホストに転送する．メインホスト上に存在しないページについては転送を行わず，そのページのアドレスなどの情報も転送しない．その代わりに，移送元サブホストのメモリサーバと通信してマイグレーションの指示を送り，サブホスト上に存在する VM のページはメモリサーバに転送させる．

サブホスト上のメモリを移送先ホストへ転送できるようにメモリサーバの拡張を行った．移送元サブホストのメモリサーバはメインホストからのマイグレーションの指示を受信すると，ページ配列に基いてサブホストに存在する VM のメモリのアドレスとデータの組を移送先ホストに転送する．メモリサーバはマイグレーション中にもメインホストとの間でリモートページングを行わなければならないため，移送先ホストへメモリを転送する機能はスレッドを用いて作成し，メモリの転送とリモートページングを並列に動作させられるようにした．その結果，ページ配列が 2 つのスレッドに同時にアクセスされる可能性があるため，ページ配列の各エントリについてロックを獲得してからアクセスするようにした．

移送先ホストの QEMU-KVM は，移送元メインホストから受信したページについては既存の機能を用いてマイグレーション処理を行う．一方，サブホストから転送されたページについてもマイグレーション処理を行えるようにす

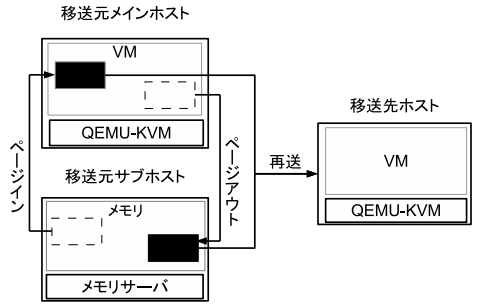

図 6 統合マイグレーション中のリモートページング

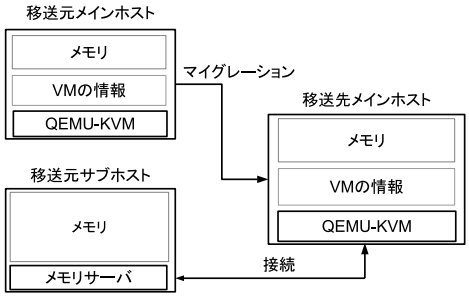

図 7 部分マイグレーションの流れ

るために，サブホストからの接続を待つスレッドを追加し，サブホストから接続されるたびにそのサブホストの処理を行うスレッドを作成するようにした．これにより，メインホストと複数のサブホストから並列にデータを受信してマイグレーション処理を行うことができる．サブホストの処理を行うスレッドでは，サブホストからメモリの物理アドレスを受信した際にそれをホストアドレスへ変換する．続いて，メモリデータを受信するとそのホストアドレスへデータを書き込む．

移送先ホストの QEMU-KVM は移送元のメインホストとサブホストからすべてのメモリデータを受信するのを待って，VM の実行を再開する．メインホストのメモリ転送が完了した時点ではサブホストからのメモリ転送が完了しているとは限らず，VM の実行が再開されてしまうと正常な動作が保証できなくなるためである．そこで，サブホストのメモリサーバがメモリの転送を完了して移送先ホストとの接続を切断するまで，移送先ホストの QEMU-KVM は VM の実行再開を待つようにした．

4.3 統合マイグレーション中のリモートページング

統合マイグレーション中にリモートページングが発生した場合でも過不足なくページを転送できるようにするために，IPmigrate では各ページの転送情報と更新情報を管理する．転送情報は，ページがすでに移送先ホストへ転送されているかどうかを表し，移送元のメインホストとサブホストでビットマップを作成して管理する．移送先ホストへメモリを転送した時には，当該メモリに対応するビットを 1 にセットすることで転送済みとする．更新情報は，移送元メインホストでページが更新されたかどうかを表し，QEMU-KVM のダーティビットマップを利用する．ページの更新が行われた時にはダーティビットマップの対応するビットが 1 にセットされる．

図 6 のように，マイグレーション中に移送元メインホストへのページインが発生した場合には，サブホストは当該ページのアドレスおよびデータとともに転送情報を移送元メインホストに送信する．移送元メインホストでは，受信した転送情報に応じて当該ページの転送処理を行う．ページインされたページが転送済みであるならば転送は行わず，未転送であるならばダーティビットマップの対応するビットに 1 をセットすることで QEMU-KVM の再送機構を用いて移送先ホストへ転送する．現在の実装ではページインされたページは必ず再送される．

一方，マイグレーション中にサブホストへのページアウトが発生した場合には，移送元メインホストは当該ページのアドレスおよびデータとともに転送情報をサブホストへ送信する．この際に，メインホストでの更新情報を転送情報に反映し，ページが更新されていれば転送情報を未転送状態に変更する．サブホストでは受信した転送情報に応じて転送処理を行う．ページアウトされたページが転送済みであればサブホストでは転送を行わず，未転送である場合だけ転送を行う．現在のところ，ページアウト後の転送処理を実装中であるため，ページインと同時に行うべきページアウトは行っていない．

4.4 部分マイグレーションの流れ

部分マイグレーションは**図 7** のように行われる．複数ホストにまたがって動作する VM の内，メインホスト間でのマイグレーションに対応するために，QEMU-KVM のマイグレーション機構の拡張を行った．移送元メインホストの QEMU-KVM は，位置データベースに基づいて移送先メインホストにメモリの転送を行う．移送元メインホストに存在するページは統合マイグレーションと同様にページのアドレスとデータの組を移送先メインホストへ転送する．ページがサブホストに存在する場合は，ページのアドレスとそのサブホストの IP アドレスを移送先メインホストへ転送する．それ以外は通常のマイグレーションと同様であり，必要に応じてメモリの再送を行う．最終的に CPU やデバイスの状態を送信して，マイグレーションを完了する．

移送先メインホストの QEMU-KVM は，分割マイグレーションと同様の処理を行う．移送元メインホストからメモリデータが送られてきた場合は，対応する VM のページにそのデータを書き込み，ページがメインホストに存在する

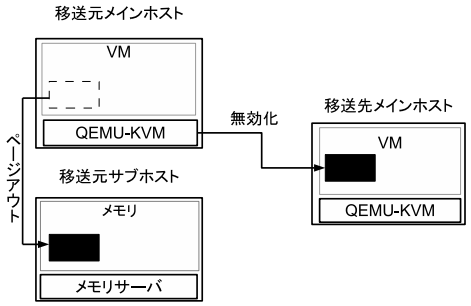

図 8 部分マイグレーション中のリモートページング

という情報を位置データベースに登録する．ページに関する情報だけが送られてきた場合には，ページがサブホストに存在するという情報を位置データベースに登録する．すべてのメモリ情報を受信したら，元のサブホストとの接続を確立して，VM の実行を再開する．

メインホスト間での部分マイグレーションを終了した後は，分割マイグレーション後と同様に Linux の userfaultfd 機構を用いて移送先メインホストと元のサブホストとの間でリモートページングを行う．VM のメモリ全体を userfaultfd 機構に登録しておくことにより，VM がメインホストに存在しないメモリにアクセスするとページフォールトが発生する．その際に QEMU-KVM へイベントが送られるため，QEMU-KVM は位置データベースを用いて当該ページがどのサブホストに存在するかを調べる．ページが存在するサブホストへページイン要求を送り，サブホストから受信したメモリデータを userfaultfd 機構を用いて VM のメモリに書き込む．同時に，VM がアクセスしないことが予測されるページを VM のメモリ割り当てから削除し，そのメモリデータをサブホストへのページアウト要求とともに送信する．

4.5 部分マイグレーション中のリモートページング

部分マイグレーション中にリモートページングが発生した場合には過不足なく，かつ，整合性を保ってメモリを転送できるようにする必要がある．そのために，移送元メインホストでは転送情報と更新情報に加えて無効化情報も管理する．無効化情報は，移送先ホストに転送済みのページを無効化する必要があることを表し，移送元メインホストにおいてビットマップを作成して管理する．サブホストからのページインが発生した場合には，移送元メインホストは必要に応じてダーティビットマップの当該ページに対応するビットをセットすることで，再送機構を用いて移送先メインホストへ転送する．

一方，**図 8** のように，移送元メインホストからのページアウトが発生した場合には，移送元メインホストは転送情報を調べ，当該ページが移送先メインホストへ転送されていた場合には，無効化情報の対応するビットを 1 にセットする．移送元メインホストは無効化情報に基づいて移送先メインホストへページの無効化を指示し，移送先ホストで VM のメモリ割り当てを削除する．この機能はページアウト処理と同様にして実現できると考えられるが，現在，実装中である．

5. 実験

IPmigrate の有効性を示すために，統合マイグレーションとメインホスト間での部分マイグレーションの性能および，マイグレーション後の VM の性能を調べる実験を行った．比較対象として，従来の 1 対 1 マイグレーションを用いた場合と分割マイグレーションを用いた場合についても調べた．VM を動作させる (メイン) ホストには，Intel Xeon E3-1270 v3 3，5GHz の CPU，16GB のメモリ，Intel X540-T2 の 10 ギガビットイーサネット (GbE) を搭載したマシンを 2 台用いた．サブホストには，Intel Xeon E3-1270v2 3.5GHz の CPU，12GB のメモリ，Intel X540-T2 の 10GbE を搭載したマシンを 1 台用いた．これらのマシンは 10GbE スイッチで接続した．統合マイグレーション先のホストでは 1 枚の NIC の 2 つのポートに対してネットワークボンディングを設定した．ホスト OS には Linux 4.3 を用い，仮想化ソフトウェアには QEMU-KVM 2.4.1 を使用した．VM には仮想 CPU を一つ割り当て，メモリは 2〜12GB 割り当てた．VM をメインホストと 1 台のサブホストに分割する場合にはメモリ全体の半分ずつになるように分割した．

5.1 マイグレーション性能

IPmigrate のマイグレーション性能を調べるために，マイグレーション時間とダウンタイムを測定した．マイグレーション時間は**図 9** のようになり，いずれも VM のメモリ量に比例した時間がかかった．統合マイグレーションを，従来のマイグレーションと比較するとマイグレーション時間は若干短くなっているものの，メインホストとサブホストからの並列転送による高速化はほぼ見られなかった．これは，X540-T2 NIC の 2 つのポートを束ねても 20Gbps の帯域は得られず，マイグレーション先のネットワーク性能がボトルネックになったためと考えられる．メインホスト間での部分マイグレーションはメモリ転送量が半分程度であるため，従来のマイグレーションの半分程度の時間となった．

ダウンタイムは**図 10** のようになり，2GB の時を除いて VM のメモリ量に関わらずほぼ一定であることが分かった．従来のマイグレーションと比較して，統合マイグレーションでは最大 63%ダウンタイムが減少し，部分マイグレーションでは最大 70%ダウンタイムが減少した．これはマイグレーションの最終段階で VM を停止させた時に転送すべ

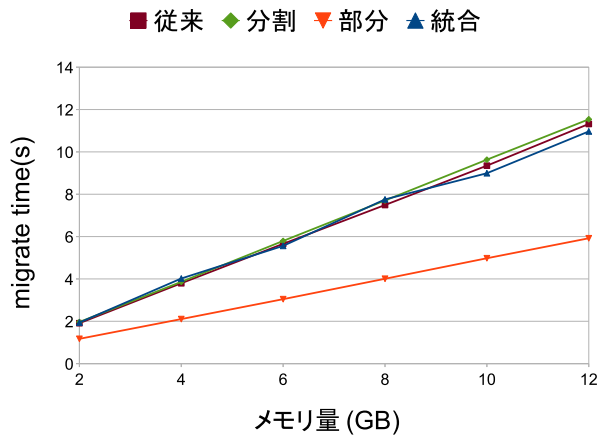

図 9　マイグレーション時間

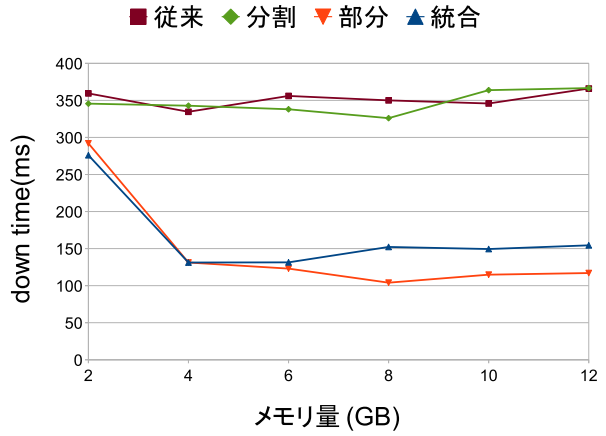

図 10　ダウンタイム

き残りのメモリページ数が異なるためと考えられる．統合マイグレーションや部分マイグレーションでは移送元メインホストに存在しないメモリがあるため，その分だけ実際に転送されるメモリページ数が減ることになる．統合マイグレーションと部分マイグレーションにおいて，VM のメモリ量が 2GB の時だけダウンタイムが長い原因は調査中である．

5.2　並列転送によるマイグレーションの高速化

統合マイグレーションにおいてメインホストとサブホストからの並列転送によるマイグレーションの高速化を確認するための実験を行った．この実験では，移送先ホストに GbE の NIC を 2 枚用いてネットワークボンディングを設定することでネットワークのボトルネックを解消した．VM に 2GB のメモリを割り当てて測定した結果，図 **11** のようになった．マイグレーション時間は従来のマイグレーションと比較して 46%減少しており，メモリの並列転送による高速化が確認できた．ダウンタイムは図 **12** のように

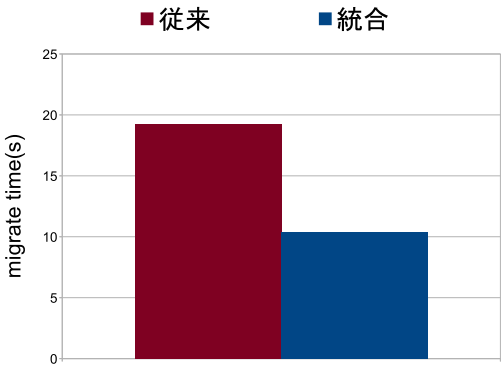

図 11　並列転送時のマイグレーション時間

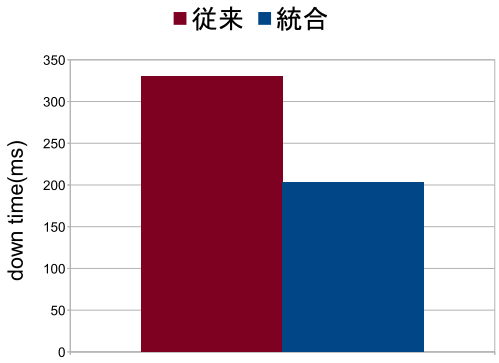

図 12　並列転送時のダウンタイム

なり，従来のマイグレーションと比較して 38%減少した．これは並列転送によるものではなく， 5.1 節と同様の理由によるものと考えられる．

5.3　マイグレーション後の VM の性能

マイグレーション後の VM の性能を調べるために，VM 内でインメモリ・データベースの memcached[4] を動作させ，memaslap ベンチマークを用いて性能を測定した．memaslap の set と get の比率は 0.6 対 0.4 に設定し，memcached が使用するメモリ量を 6GB とした．この実験では，VM に 12GB のメモリを割り当て，各ホストは 10GbE を用いて接続した．図 **13** に示す実験結果から，従来のマイグレーション後と比べて，分割マイグレーション後には 11%の性能低下が見られた．しかし，統合マイグレーション後にはほぼ性能低下は見られなかった．これは，リモートページングによる性能低下が解消されたためである．部分マイグレーション後は分割マイグレーション後と同じシステム構成となるため，同等の性能となった．

6.　関連研究

ポストコピーマイグレーション [5] は，CPU の状態などの VM を実行する上で必要な情報だけを移送先ホストへ転

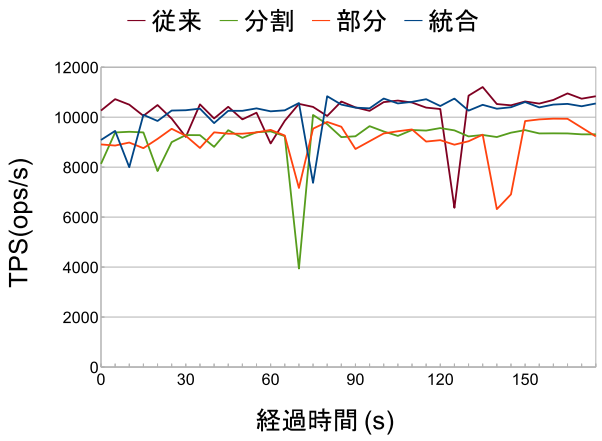

図 13　マイグレーション後の memcached の性能

送した後，すぐに移送先ホストで VM の実行を再開する手法である．移送元ホストに残っているメモリはオンデマンド転送かバックグラウンド転送を用いて移送先ホストへ転送する．VM の実行を移送先ホストで再開させるまでの動作は，VM 本体だけを別のホストに移動させる部分マイグレーションと考えることができる．その後の動作は，サブホストのメモリをメインホストに転送する部分マイグレーションと考えることができる．

Scatter-Gather マイグレーション [6] では，マイグレーションを行う際に，移送先ホストと移送元ホストの間で複数の中間ホストを用いる．VM のメモリを移送元ホストから中間ホストへ高速に転送することで，移送元ホストを停止させられるようになるまでの時間を短くすることができる．移送元ホストでは，オンデマンド転送やバックグラウンド転送を用いて中間ホストから VM のメモリを取得する．移送元ホストから複数の中間ホストへメモリを転送する際の動作は分割マイグレーションと類似しており，複数の中間ホストから移送先ホストへメモリを転送する際の動作は統合マイグレーションに類似している．ただし，Scatter-Gater マイグレーションでは VM 本体はマイグレーションの初期段階で移送元ホストに移動しているため，マイグレーション中のページイン処理は単純なオンデマンド転送となる．また，ページアウト処理を行う必要はない．

MemX[8] では，VM の起動時から複数のホストのメモリを利用可能であり，分割マイグレーション後と同じ状態である．MemX-VM モードでは，VM 内のゲスト OS が提供するブロックデバイス経由で MemX サーバのメモリへのアクセスを行う．VM のマイグレーションを行う際に MemX サーバのメモリの転送を行わない点で，メインホスト間での部分マイグレーションと考えることができる．リモートページングはゲスト OS が行うため，マイグレーション中にリモートページングが発生してもメモリを過不足なく転送することができる．一方，Xen のドメイン 0 でブロックデバイスを提供する MemX-DD モードや VM に拡張メモリを提供する MemX-VMM モードではマイグレーションには対応していない．

MemX は MemX サーバのメモリを別の MemX サーバに転送するページマイグレーションもサポートしている．これはサブホスト間での部分マイグレーションと考えることができる．しかし，マイグレーション中のリモートページングについては考慮されておらず，評価も行われていない．また，MemX サーバを VM 内で動作させ，VM ごと MemX サーバとそのメモリをマイグレーションすることも提案されている．ただし，この手法を用いるとリモートページングのオーバヘッドが増大する可能性が指摘されている．

Agile ライブマイグレーション [9] では，スワップデバイスをネットワーク上に配置し，移送元ホストに存在するメモリのみを移送先ホストへ転送する．これは，メインホスト間での部分マイグレーションに類似している．Agile ライブマイグレーションでは，スワップデバイスにできるだけ多くのメモリをページアウトしておくことでマイグレーション時に転送するメモリを少なくすることができる．そのために，VM のワーキングセットを追跡して VM が必要とするメモリ以外をページアウトする．しかし，Agile ライブマイグレーションではマイグレーション中のページングを考慮していない．

Jettison [10] は電力消費を削減するためにデスクトップ VM の部分マイグレーションを行う．Jettison では，使われていないデスクトップ VM のワーキングセットメモリだけをサーバに転送して高速にデスクトップ VM を集約し，デスクトップの電力消費を抑える．デスクトップ VM が使われ始めると再びデスクトップで実行されるようにマイグレーションを行う．

vNUMA[7] では，複数のホストのメモリや CPU を用いて一つの VM の動作を可能にする．これにより，1 台のホストではリソースが不足していても複数のホストを用いて大容量メモリを持つ VM を動作させることができる．vNUMA では，分散共有メモリを用いることで，メモリが存在するホストを気にすることなくメモリのアクセスを行うことができる．しかし，vNUMA はマイグレーションには未対応である．

7. まとめ

本稿では，複数ホストにまたがって動作している VM のマイグレーションを可能にするシステム IPmigrate を提案した．IPmigrate は，複数ホストに分割された VM を一つに統合する統合マイグレーションと，分割された VM の内の一部だけを別ホストに移動する部分マイグレーションを可能にする．統合マイグレーションでは，移送元のメイン

ホストとサブホストから並列にメモリを転送することでマイグレーションの高速化を図る．これらのマイグレーション中にリモートページングが発生した場合には，IPmigrate は当該ページの再送や無効化を行うことで過不足なくメモリを転送する．IPmigrate を KVM に実装し，統合マイグレーションとメインホスト間での部分マイグレーションを実現した．実験により，統合マイグレーションにおけるメモリの並列転送の効果とリモートページングによる性能低下の解消を確認した．

今後の課題は，マイグレーション中にページアウトされたメモリを過不足なく転送するための実装を完了させることである．また，統合マイグレーションでサブホストからのメモリ転送に時間がかかる場合のダウンタイム削減も今後の課題である．現在の実装では，移送先ホストで同期をとっているため，移送元メインホストが VM を停止させた後で同期待ちを行う場合がある．そのため，移送元メインホストで同期をとるようにすることを検討している．さらには，サブホスト間での部分マイグレーションなど，より柔軟な部分マイグレーションをサポートすることも計画している．

参考文献

[1] Apache Software Foundation. Apache Spark - Lightning-Fast Cluster Computing. http://spark.apache.org/.
[2] Facebook, Inc. Presto: Distributed SQL Query Engine for Big Data. https://prestodb.io/.
[3] M. Suetake. H. Kizu. and K. Kourai. Split Migration of Large Memory Virtual Machines. In Proceedings of the 7th ACM Asia-Pacific Workshop on Systems (2016).
[4] B. Fitzpatrick. memcached - A Distributed Memory Object Caching System. http://memcached.org/.
[5] M. Hines, and K. Gopalan. Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning. In Proceedings of International Conference on Virtual Execution Environments,pp. 51–60 (2009).
[6] U. Deshpande, Y. You, D. Chan, N. Bila, and K. Gopalan. Fast Server Deprovisioning through Scatter-Gather Live Migration of Virtual Machines. In Proceedings of the 7th IEEE International Conference on Cloud Computing, pp.376–383 (2014).
[7] M. Chapman and G.Heiser. vNUMA: A Virtual Shared-Memory-Multi Processer. In Proceeding of Conference USENIX Annual Technical Conference (2009).
[8] U. Deshpande, B. Wang, S. Haque, M. Hined, and K. Gopalan. MemX: Virtualization of Cluster-Wide Memory. In Proceedings of International Conference on Parallel Processing, pp.663–672 (2010).
[9] U. Deshpande, D. Chan, T. Guh, J. Edouard, K. Gopalan, N. Bila, Agile Live Migration of Virtual Machines, In Proceedings of International Parallel and Distributed Processing Symposium (2016).
[10] N. Bila, E. Lara, K. Joshi, H. Lagar-Cavilla, M. Hiltunen, and M. Satyanarayanan. Jettison: Efficient Idle Desktop Consolidation with Partial VM Migration. In Proceedings of the 7th ACM European Conference on Computer Systems, pp. 211-224 (2012).