分割メモリVMの高速かつ柔軟な部分マイグレーション

柏木　崇広[1,a)]　末竹　将人[1]　光来　健一[1]

概要：近年，IaaS 型クラウドでは大容量メモリを持つ仮想マシン（VM）が提供されるようになってきた．このような VM のマイグレーションを容易にするために，VM のメモリを分割して複数ホストに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，マイグレーション後に複数ホストにまたがって動作する VM（分割メモリ VM）をマイグレーションすると，リモートページングが頻発するためオーバヘッドが大きい．また，分割メモリ VM が動作しているホストの一部をメンテナンスする場合でも VM 全体をマイグレーションする必要がある．本稿では，分割メモリ VM に対して高速かつ柔軟に部分マイグレーションを行う IPmigrate を提案する．IPmigrate は VM の一部をホスト単位でマイグレーションすることにより，一部のホストのメンテナンスを可能にする．また，分割メモリ VM が動作している各ホストから直接，一つのホストにマイグレーションすることにより，VM を効率よく再統合することができる．部分マイグレーション中に VM がリモートページングを発生させた場合には当該ページの再送や無効化を行うことにより，VM の整合性を保つ．我々は IPmigrate を KVM に実装し，部分マイグレーションの有用性を示す実験を行った．

1. はじめに

近年，IaaS 型クラウドでは大容量メモリを持つ仮想マシン（VM）も提供されるようになってきている．例えば，Amazon EC2 では 4TB 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を行うことが可能となる [1][2]．VM を用いる一つの利点として，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り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を用意するのが従来よりも難しくなる．

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 CPU やデバイスの状態等の 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これらのホストにまたがって動作する VM は**分割メモリ** *VM* と呼ばれ，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いながら動作する．しかし，従来のマイグレーションを用いて分割メモリ VM をマイグレーションすると，リモートページングが頻発してマイグレーション性能が大幅に低下する．また，一部のホストをメンテナンスする際でも VM 全体をマイグレーションしなければならない．

そこで本稿では，分割メモリ VM に対して高速かつ柔軟な部分マイグレーションを可能にする *IPmigrate* を提案する．IPmigrate では，移送元の各ホストが必要に応じて分割メモリ VM の一部を移送先ホストに直接転送する．分割メモリ VM の一部をホスト単位で転送するマイグレーションは置換マイグレーションと呼ばれ，一部のホストのメンテナンスが可能になる．また，分割メモリ VM を一つのホストに集約するマイグレーションは統合マイグレーションと呼ばれ，各ホストから VM のメモリを直接転送することで効率のよいマイグレーションが実現できる．

我々は，IPmigrate を KVM に実装し，メインホストとサブホストの置換マイグレーションと統合マイグレーションを実現した．置換マイグレーションはメインホストまたはサブホストに存在する VM のメモリだけを転送し，マイグレーション中に VM がリモートページングを発生させた場合には再送もしくは無効化を行う．それにより，すべてのメモリが過不足なく，整合性を保ちながら転送されることを保証する．統合マイグレーションはメインホストとサブホストの置換マイグレーションを組み合わせて実現されるが，整合性や効率の点から専用の機構を実装した．実験

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

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 1 対 1 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 章で分割メモリ VM の高速かつ柔軟な部分マイグレーションを可能にする *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．転送中に更新された 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少なくなったら移送元ホストの VM を停止させる．そして，VM の更新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 4TB 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で効率が悪いためである．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．

2.2 分割マイグレーション

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転送する分割マイグレーション [3] が提案されている．マイグレーション後に CPU やデバイスの状態などの VM コアを動作させるホストはメインホスト，メインホスト以外のホストはサブホストと呼ばれる．分割マイグレーションは VM コア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群に転送する．マイグレーション後に VM がアクセスするメモリはアクセス履歴に基づいて決定する．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，効率よくマイグレーションを行うことができる．

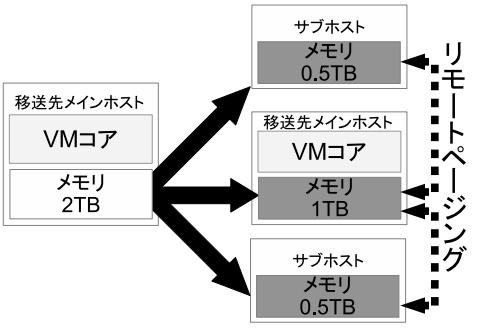

図 1 分割マイグレーション

本稿では，分割マイグレーション後に複数ホストにまたがって動作する VM を**分割メモリ** *VM* と呼び，それぞれのホスト上にある VM の一部を *VM* **フラグメント**と呼ぶ．分割メモリ VM はメインホストとサブホスト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．メインホスト上の VM コア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，今後アクセスしないことが予測されるメインホスト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後にアクセスが予測されるメモリはメインホストに転送されているため，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．

分割マイグレーション後にホストのメンテナンスや負荷分散を行う際には，分割メモリ VM をマイグレーションする必要がある．その際に，移送先ホストとして十分な空きメモリを持つホストが確保できなかった場合には，再び分割マイグレーションが行われる．一方，十分な空きメモリを持つホストが準備できた際には，メンテナンスや負荷分散のためでなくとも分割メモリ VM をマイグレーションして一つのホストで動作させるのが望ましい．これはリモートページングによる性能低下を解消するためである．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，リモートページングは VM の性能に対して影響が大きい．

しかし，従来の 1 対 1 マイグレーションや分割マイグレーションを用いて分割メモリ VM をマイグレーションするといくつかの問題が生じる．第一に，マイグレーション中にリモートページングが頻発することにより，マイグレーション性能が大幅に低下する．従来のマイグレーションはメインホストからのみ移送先ホストにメモリを転送するため，サブホストに存在するメモリを転送する際には一度，当該メモリをメインメモリにページインする必要がある．我々の実験によると，分割メモリ VM に従来の 1 対 1 マイグレーションを適用すると，一つのホストで動作する VM のマイグレーションと比較して 6 倍の時間がかかることが分かった．

表 1　部分マイグレーションの分類（A, B, C は移送元ホスト，P, Q は新しいホスト，*は VM コアが存在するホスト，a,b,c,d はメモリ領域）

種類	マイグレーション前	マイグレーション後
置換	A*[a], B[b], ...	P*[a], B[b], ... A*[a], P[b], ...
統合	A*[a], B[b], ...	P*[ab], ... A*[ab], ... B*[ab], ...
	A*[a], B[b], C[c], ...	A*[a], P[bc], ... A*[a], B[bc], ...
分割	A*[ab], ...	P*[a], Q[b], ... A*[a], P[b], ... P*[a], A[b], ...
	A*[a], B[bc], ...	A*[a], P[b], Q[c], ... A*[a], B[b], P[c], ...
分割＆統合（例）	A*[a], B[bc], ...	A*[ab], P[c], ... A*[ab], B[c], ...
	A*[a], B[b], C[cd], ...	A*[a], B[bc], P[d], ... A*[ac], B[bd], ...

第二に，従来のマイグレーションでは分割メモリ VM 全体をマイグレーションすることしかできない．分割メモリ VM が動作している複数のホストの内，一部のホストだけをメンテナンスする場合や一部のホストだけ負荷が高い場合にも，それ以外のホスト上にある VM フラグメントを別のホストに転送しなければならない．これでは，分割メモリ VM のマイグレーションする必要のない部分までもマイグレーションするため非効率である．

3. 部分マイグレーションの分類

本稿では，分割メモリ VM に対して柔軟かつ高速な部分マイグレーションを可能にする IPmigrate を提案する．

3.1 分割メモリ VM の部分マイグレーション

IPmigrate の部分マイグレーションは移送元のホスト群で動作している分割メモリ VM の一部または全体を移送先のホスト群に移動させる．移送元と移送先のホスト群はすべてのホストが異なる場合，一部が異なる場合，すべて同じ場合がありえる．このような部分マイグレーションの際に，IPmigrate では移送元の各ホストが必要に応じて分割メモリ VM の一部を移送先のホストに直接転送する．リモートページングを行って移送元メインホストから移送先ホストに転送する必要がないため，高速なマイグレーションが可能になる．考えられる部分マイグレーションは表 1 にように分類される．

置換マイグレーションは，対象ホスト上にある VM フラグメントを別のホストに転送する部分マイグレーションである．メインホスト上の VM コアとメモリを転送する場合と，サブホスト上の VM のメモリを転送する場合が考えられる．置換マイグレーションは分割メモリ VM が動作している複数のホストの内，一部のホストだけのメンテナンスを行う際に高速なマイグレーションを可能にする．

統合マイグレーションは，いくつかのホスト上にある VM フラグメントを一つのホストに統合する部分マイグレーションである．メインホストとサブホストを統合する場合とサブホスト同士を統合する場合が考えられる．より大きな空きメモリを持つホストが準備できた場合にはそのホストを移送先ホストとして統合を行うが，分割メモリ VM が動作しているいずれかのホストのメモリに空きができた場合にはそのホストを移送先として統合することもできる．統合マイグレーションはメインホストにより多くの VM フラグメントを集約してリモートページングを減らしたり，性能の高いサブホストに VM フラグメントを集約したりするのに有用である．

分割マイグレーションは，一つのホスト上にある VM フラグメントを複数のホストに分割する部分マイグレーションである．メインホスト上の VM フラグメントを分割する場合とサブホスト上の VM フラグメントを分割する場合が考えられる．分割メモリ VM が動作しているホストのメンテナンスを行う際に移送先として十分な空きメモリを持つホストが確保できない場合には，そのホスト上にある VM フラグメントを複数の新しいホストに分割する．一方，分割メモリ VM が動作しているホストの空きメモリが少なくなった場合には，そのホスト上にある VM フラグメントを新しいホストに転送することもできる．先行研究で提案されている分割マイグレーション [3] は，一つのホスト上だけで動作している VM を複数の新しいホストに分割するという特殊な場合である．

移送元では分割マイグレーションが行われ，移送先では統合マイグレーションが行われる複合的なマイグレーションも考えられる．例えば，移送元の一つのホスト上にある VM フラグメントを分割して他の複数の既存ホストに統合したり，一部だけ新しいホストに転送したりすることができる．また，メインホストとサブホスト間やサブホスト間で VM フラグメントの一部だけを移動させることもできる．

以下，本稿では置換マイグレーションと一つの新しいホストへの統合マイグレーションについて詳しく説明する．

3.2 置換マイグレーション

メインホストの置換マイグレーションを行う場合，**図 2** のように移送元メインホストに存在する VM コアおよびメモリ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．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サブホストとの間でリモートページングを行いながら分割メモリ VM の実行を行う．一方，サブホスト

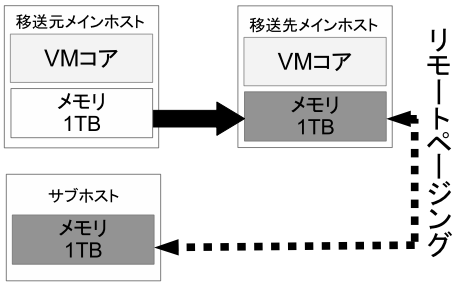

図 2 メインホストの置換マイグレーション

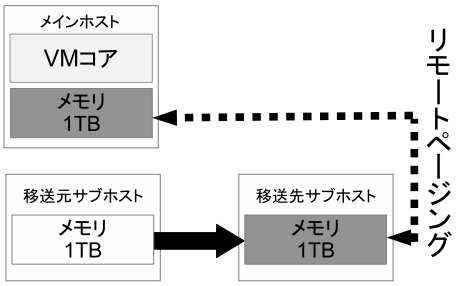

図 3 サブホストの置換マイグレーション

の置換マイグレーションを行う場合，**図 3** のように移送元サブホストに存在する VM のメモリのみを移送先サブホストへ転送する．マイグレーションが完了すると，メインホストは移送先サブホストとの間でリモートページングを行いながら分割メモリ VM の実行を行う．

置換マイグレーション中に分割メモリ VM がサブホストのメモリにアクセスすることにより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．メインホストの置換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すると，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そのページを転送する．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，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すると，当該ページを移送先メインホストへ転送済みであれば，移送先メインホストで当該ページの無効化を行う．これは，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である．

サブホストの置換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も，IPmigrate は同様の処理を行う．メインホストからのページアウトが発生すると，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．メインホストへのページインが発生すると，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．

3.3 統合マイグレーション

統合マイグレーションはメインホストとサブホストのそれぞれの置換マイグレーションを組み合わせ，移送先ホストとして同一のホストを指定することで実現することができる．ただし，これら 2 つの置換マイグレーションを単純に組み合わせるだけでは，整合性を保ちつつ効率よく統合マイグレーションを行うことはできない．複数の移送元ホストから並列に置換マイグレーションを行うことにより，高速な統合マイグレーションを実現する．そのため

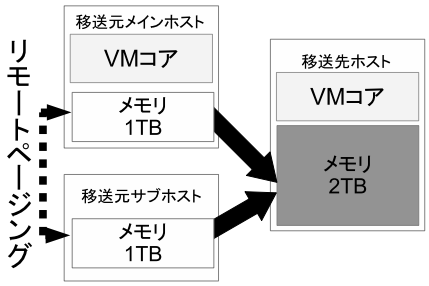

図 4 統合マイグレーション

に，ネットワークおよび移送先ホストにおいて並列転送をサポートする．

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には，IPmigrate は転送すべきページであるかどうかの情報を付与してページイン・ページアウトを行う．統合マイグレーションではそれぞれの移送元ホストで独立にメモリが転送されるため，転送情報を移送元ホスト間で受け渡す必要がある．また，転送済みのページに対してリモートページングが発生しても，移送先ホストにおいて当該ページの無効化は行わない．すべてのページは最終的に移送先ホストに転送されるため，単一ホストの置換マイグレーションのように複数のホストに同一ページが存在する状態になることはない．

4. 実装

我々は IPmigrate を QEMU-KVM 2.4.1 および Linux 4.3 に実装した．

4.1 システム構成

IPmigrate のシステム構成は**図 5** のようになる．メインホストでは IPmigrate を実装した QEMU-KVM を動作させ，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．QEMU-KVM 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**ネット**

図 5 IPmigrate のシステム構成

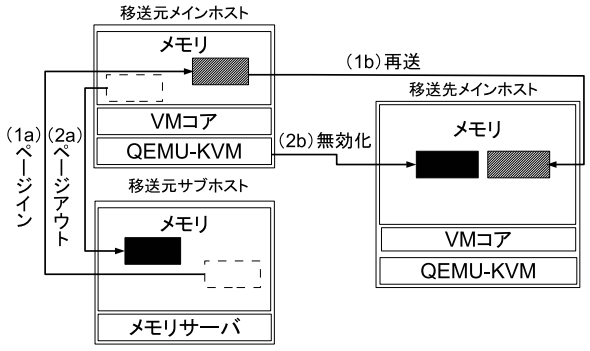

図 6 メインホストの置換マイグレーション中のページング処理

ワーク・ページテーブルを管理する．ネットワーク・ページテーブルはページフレーム番号からホスト ID を検索するための表と，ホスト ID から IP アドレスを検索するための表からなり，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ネットワーク・ページテーブルに登録される．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．

メモリサーバは VM のメモリのどの部分がそのサブホストに存在するかという情報が登録されたページ・サブテーブルを管理する．ページ・サブテーブルはページフレーム番号からメモリデータを検索するための表であり，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・サブテーブルから削除される．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・サブテーブルに登録される．

部分マイグレーションの移送元となるホストでは，転送ビットマップ，再送ビットマップ，無効化ビットマップの３つの情報を管理する．転送ビットマップはページを移送先ホストに転送済みであるかどうかを管理し，転送済みであれば対応するビットがセットされる．再送ビットマップはページを再送すべきかどうかを管理し，再送すべきであれば対応するビットがセットされる．メインホストでは既存のダーティビットマップを利用する．無効化ビットマップは移送先ホストのページを削除すべきかどうかを管理し，削除すべきであれば対応するビットがセットされる．

4.2 メインホストの置換マイグレーション

4.2.1 マイグレーションの流れ

メインホストの置換マイグレーションでは，移送元メインホストの QEMU-KVM がネットワーク・ページテーブル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については物理アドレスとデータの組を転送し，転送ビットマップの対応するビットをセットする．ページがサブホストに存在する場合は，その物理アドレスとサブホストの IP アドレスだけを移送先メインホストへ転送する．それ以外は従来の 1 対 1 マイグレーションと同様であり，マイグレーション中に VM によって書き換えられたページについては再送を行う．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，マイグレーションの開始時に VM のメモリ全体を Linux の userfaultfd 機構に登録しておく．userfaultfd 機構はページフォールトが発生した際にその情報を QEMU-KVM に通知するための機構である．移送元メインホストからメモリデータが送られてくると，対応する VM のページに userfaultfd 機構を用いて書き込み，ページがメインホストに存在するという情報をネットワーク・ページテーブルに登録する．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ネットワーク・ページテーブルに登録する．すべてのメモリ情報と VM コアを受信したら，移送元サブホストとの接続を確立して，VM の実行を再開する．置換マイグレーションが完了した後は，分割マイグレーション後と同様に移送先メインホストと移送元のサブホストとの間でリモートページングを行う．

4.2.2 リモートページングへの対処

メインホストの置換マイグレーション中に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において転送ビットマップを調べる．当該ページが未転送であれば再送ビットマップの対応するビットをセットする．その結果，QEMU-KVM の再送機構により，そのページは移送先メインホストへ転送される．一方，**図 6** 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も，移送元メインホストは転送ビットマップを調べる．ページインの場合と異なり，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，無効化ビットマップの対応するビットをセットする．移送元メインホストのマイグレーション用スレッドは無効化ビットマップに基づいて

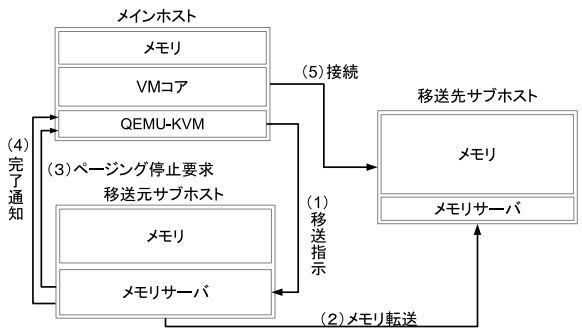

図 7　サブホストの置換マイグレーションの手順

移送先メインホストへページの無効化要求を送信し，移送先メインホストで VM のメモリから当該ページを削除する．ページの削除は userfaultfd 機構の拡張機能 [3] を用いて行う．

4.3　サブホストの置換マイグレーション

4.3.1　マイグレーションの流れ

サブホストの置換マイグレーションを行う際には，まず，メインホストに要求を送信する．要求を受け取ったメインホストは**図 11** のように移送元サブホストに置換マイグレーションの要求を送信する．移送元サブホストはページ・サブテーブルに基づいてサブホストに存在するページの物理アドレスとデータの組を移送先サブホストへ転送し，転送ビットマップの対応するビットをセットする．移送元サブホストの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，リモートページングと並列に動作させる．メインホストからのページアウトによって再送ビットマップがセットされていた場合には当該ページの再送を行う．置換マイグレーション中のページアウトの処理については 4.3.2 節で説明する．

再送すべきページが十分に少なくなったら，メインホストにリモートページングの停止要求を送信する．この要求を受信した QEMU-KVM は，VM がページフォールトを発生させてもサブホストにページイン要求を送らないようにロックを取得する．その後，移送元サブホストがすべてのページを転送し終えるとメインホストにマイグレーション完了の通知を送る．通知を受け取ったメインホストは，ネットワーク・ページテーブルにおいて転送元ホストの ID から検索される IP アドレスを移送先サブホストのものに切り替える．最後に，移送先サブホストとの接続を確立し，リモートページングのロックを解除する．

4.3.2　リモートページングへの対処

サブホストの置換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もメインホストの置換マイグレーションと同様の処理を行う．メインホストへのページインが発生した場合には，移送元サブホストにおいて転送ビットマップを調べ，当該ページが移送先サブホストへ転送されていれば無効化ビットマップの対応するビットをセットする．移送元サブホストのマイグレーション用スレッドは無効化ビットマップに基づいて移送先サブホストへ当該ページの無効化要求を送信し，移送先サブホストで VM のメモリから削除する．メインホストからページアウトが発生した場合には，再送ビットマップの対応するビットをセットする．移送元サブホストは再送ビットマップに基づいて移送先サブホストへ当該ページを転送する．

4.4　統合マイグレーション

4.4.1　マイグレーションの流れ

統合マイグレーションでは，メインホストの置換マイグレーションと並行して，同じ移送先ホストに対してサブホストの置換マイグレーションを実行する．メインホスト単独の置換マイグレーションと異なり，移送元メインホストはサブホストに存在するページについては移送先ホストにページの情報を転送しない．移送元サブホストのページも移送先ホストに転送されるためである．

移送先ホストの QEMU-KVM は，移送元メインホストから転送されたページについてはメインホストの置換マイグレーションと同様の処理を行う．一方，移送元サブホストから転送されたページについてはサブホストからの接続を待つスレッドを用意し，接続されるたびにそのサブホストの置換マイグレーションを処理するスレッドを作成する．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メモリを受信してマイグレーション処理を行う．サブホストの処理を行うスレッドでは，受信したメモリデータを userfaultfd 機構を用いて対応する VM のページに書き込む．

複数の移送元ホストから一つの移送先ホストにメモリを転送する際に，移送先ホストで別々の NIC を用いることで高速な並列転送を可能にする．そのために，NIC ごとにルーティングテーブルを作成し，NIC に割り振られた IP アドレスを基にそれらのルーティングテーブルを参照するためのルールを追加する．このようにルーティングを設定しない場合は同一セグメントでは一つの NIC しか使うことができない．複数 NIC をボンディングで束ねる方法も考えられるが，リンクアグリゲーションを設定したネットワークスイッチが一つの NIC にのみデータを転送するようになっている場合には複数の NIC を活用できない．

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のどちらのメモリ転送が先に完了するかによって処理が異なる．移送元メインホストからのメモリ転送が先に完了した場合は，メインホスト単独のマイグレーションの場合と同様にマイグレーションを完了させる．そ

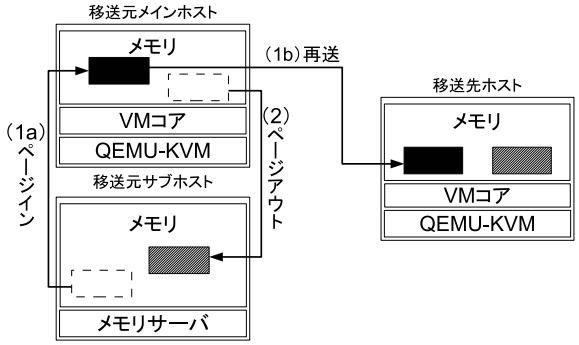

図 8　統合マイグレーション中のページング処理

の際に，移送元サブホストにおけるメモリの転送状況は考慮しない．移送先ホストでは移送元サブホストとの接続を確立し，リモートページングを行いながら VM を動作させる．その後は，サブホスト単独の置換マイグレーションと同様となり，移送元サブホストは引き続き，移送先ホストへのメモリ転送を行う．サブホストに存在するメモリをすべて転送すると，移送先ホストとの接続を切断して統合マイグレーションを完了させる．

移送元サブホストからのメモリ転送が先に完了した場合には，移送元サブホストではマイグレーションを完了させず，メモリサーバを待機させる．移送元メインホストからページング要求を受信するとリモートページングの処理を行い，必要であれば移送先ホストにそのページの転送を行う．移送元メインホストでは引き続き，移送先ホストへのメモリ転送を行い，マイグレーションの最終段階で VM を停止させてから移送元サブホストへマイグレーション完了の要求を送信する．移送元サブホストは要求を受け取るとメモリサーバを停止させてマイグレーションを完了させる．この場合も，移送元サブホストのマイグレーションを先に完了させるのが望ましいが，移送先ホストの QEMU-KVM にメモリサーバの役割をさせる必要があるため，今後の課題となっている．

4.4.2　リモートページングへの対処

図 8 のように，統合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，移送元サブホストは当該ページの転送情報も一緒に移送元メインホストに送信する．その際に，再送ビットマップで対応するビットがセットされている場合は未転送とみなす．移送元メインホストでは受信した転送情報に応じて，当該ページが転送済みであるならば転送も無効化も行わず，未転送であるならば再送ビットマップの対応するビットをセットすることで移送先ホストへ転送する．

統合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も同様に，移送元メインホストは当該ページの転送情報も一緒にサブホストへ送信する．この際に，メインホストで当該ページが更新されていれば未転送とみなす．サブホストでは受信した転送情報に応じて，当該ページが転送済みであれば転送も無効化も行わず，未転送の場合だけ転送を行う．

メインホストの置換マイグレーションが先に完了した場合，移送先ホストと移送元サブホストの間でもリモートページングが発生する．移送先ホストからページイン要求があると，移送元サブホストは当該ページを転送する．ただし，その際に，移送先ホストは十分なメモリを持っているため，ページアウトは行わない．

5.　実験

IPmigrate の有効性を示すために，メインホストとサブホストの置換マイグレーションと統合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．比較として，一つのホストで動作する VM に対して，従来の 1 対 1 マイグレーションを行った場合と複数ホストへの分割マイグレーションを行っ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 の CPU，32GB 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 E3-1270 v2 の CPU，16GB 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 つの NIC の 2 つのポートに対してルーティングの設定を行った．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 1 個割り当て，メモリは 4〜24GB 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 VM のメモリ全体の半分ずつになるように分割した．

5.1　分割メモリ VM の 1 対 1 マイグレーション性能

二つのホストにまたがって動作する分割メモリ VM 全体をマイグレーションする際に，従来の 1 対 1 マイグレーションを用いた場合のマイグレーション時間とマイグレーション中のページング量を測定した．比較として，統合マイグレーションを用いた場合の性能も測定した．この実験では，VM のメモリ量を 24GB とした．マイグレーション時間は図 **9** のようになり，1 対 1 マイグレーションを用いると大量のリモートページングが発生し統合マイグレーションの 12 倍の時間がかかった．マイグレーション中のページング量は図 **10** のようになり，1 対 1 マイグレーションを用いると統合マイグレーションの 154 倍のページング量となった．

5.2　部分マイグレーション性能

IPmigrate による部分マイグレーションの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．

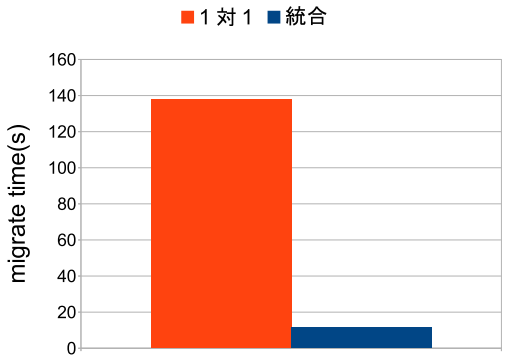

図 9　分割メモリ VM のマイグレーション時間

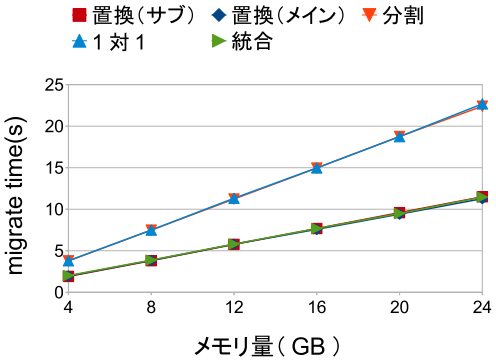

図 11　部分マイグレーションにかかる時間

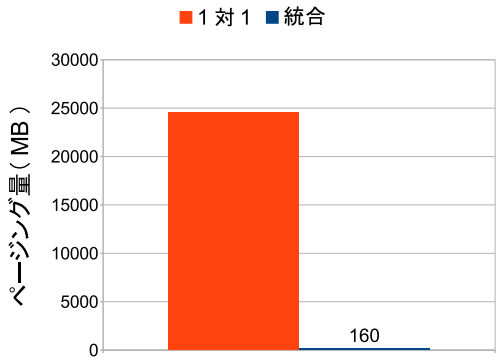

図 10　マイグレーション中のページング量

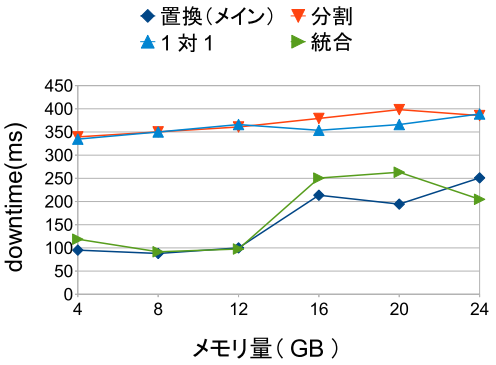

図 12　部分マイグレーション中のダウンタイム

マイグレーション時間は図 11 のようになり，いずれのマイグレーションでも VM のメモリ量に比例した時間がかかった．置換マイグレーションでは VM の半分のメモリだけを転送したため，1 対 1 マイグレーションや分割マイグレーションと比べてマイグレーション時間が約半分となった．統合マイグレーションでは VM のメモリ全体が転送されたが，マイグレーション時間は 1 対 1 マイグレーションの半分程度となった．これは，メインホストとサブホストから VM のメモリを並列転送することにより高速化することができたためである．

ダウンタイムは**図 12** のようになった．サブホストの置換マイグレーションは最終段階で VM を停止させないため，測定を行っていない．QEMU-KVM では 300 ミリ秒以下で残りのメモリを転送できると予測した時に VM を停止させてマイグレーションの最終段階に入る．1 対 1 マイグレーションと分割マイグレーションでは 300 ミリ秒に近いダウンダイムとなったが，置換マイグレーションと統合マイグレーションでは最短で 100 ミリ秒程度に減少した．これはマイグレーションの最終段階におけるメモリ転送時間が予測より短かかったためである．移送元メインホストに存在しないメモリを含めて転送時間を予測したため，実際に転送されたメモリページ数が少なくなったと考えられる．VM のメモリ量が 14GB 以上の場合にダウンタイムが増加する原因は調査中である．

5.3　高負荷 VM の部分マイグレーションの性能

VM の高負荷時のマイグレーション性能を調べるために，VM 内でメモリの書き換えを継続的に行うプログラムを動作させながらマイグレーションを行った．この実験では VM に 24GB のメモリを割り当て，6GB のメモリを書き換えながら部分マイグレーションを行った．マイグレーション時間は**図 13** のようになり，サブホストの置換マイグレーションを除いて低負荷時よりマイグレーション時間が増加した．これはメモリへの書き込みによって大量の再送が発生したためである．サブホストの置換マイグレーションでマイグレーション時間が増加しなかったのは，ワーキングセットがメインホストの利用可能なメモリより小さく，サブホストに更新されたメモリがページアウトされなかったためである．分割マイグレーションにおいてマイグレーション時間の増加率が小さい原因は現在調査中である．

ダウンタイムは**図 14** のようになり，いずれのマイグレーションでもダウンタイムは増加した．1 対 1 マイグレーションと分割マイグレーションでダウンタイムが増加したのは，マイグレーションの最終段階で転送するメモリ

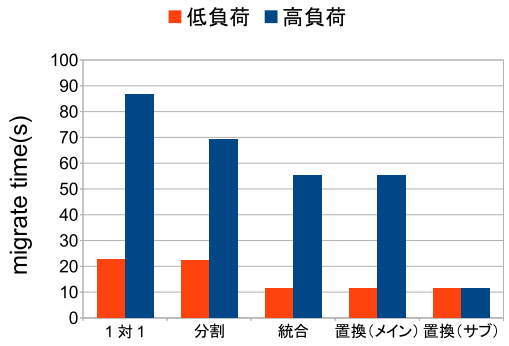

図 13　高負荷 VM の部分マイグレーションにかかる時間

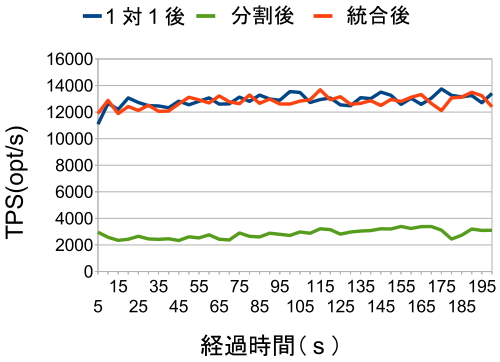

図 15　マイグレーション後の memcached の性能

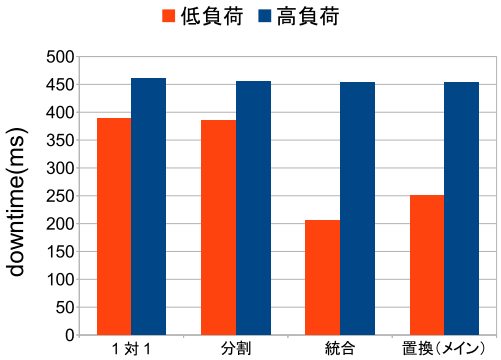

図 14　高負荷 VM の部分マイグレーション中のダウンタイム

量が増加したためと考えられる．統合マイグレーションとメインホストの置換マイグレーションでダウンタイムが増加したのは，再送されるページがすべてメインホストにあり，残りのメモリ転送にかかる時間が予測と一致したためである．

5.4　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．マイグレーション前に set と get の比率を 1 対 0 に設定して memaslap を 300 秒間実行しておき，マイグレーション後に比率を 0.6 対 0.4 に設定して 200 秒間実行した．この実験では VM に 24GB のメモリを割り当て，memcached が使用するメモリ量を 12GB とした．**図 15** に示す実験結果から，1 対 1 マイグレーションを行った後に比べて，分割マイグレーション後には 78%の性能低下がみられた．これは memcached のワーキングセットがメインホストで利用可能なメモリ量より大きく，リモートページングが頻発したためである．一方，この分割メモリ VM に対して統合マイグレーションを行った後にはほぼ性能低下は見られなくなった．すべてのメモリが一つの移送先ホストに転送され，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．

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 コア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移送先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 コア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似た状態である．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，メインホストの置換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．これはサブホストの置換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，性能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の置換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部分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する分割メモリ VM のマイグレーションを可能にする IPmigrate を提案した．IPmigrate は，分割メモリ VM の一部だけをホスト単位で置き換える置換マイグレーション，分割メモリ VM を各ホストから直接，一つのホストに統合する統合マイグレーション，一つのホスト上で動作する分割メモリ VM の一部を複数のホストに分割する分割マイグレーション，分割マイグレーションと統合マイグレーションを組み合わせた複合的なマイグレーションを可能にする．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には，IPmigrate は当該ページの再送や無効化を行うことで，整合性を保ちつつ過不足なくメモリ転送を行う．我々は IPmigrate を KVM に実装し，メインホストとサブホストの置換マイグレーションと一つの新しいホストへの統合マイグレーションを実現した．さらに，実験により各マイグレーションの有用性を示した．

今後の課題は，メモリの無駄な再送をまったく行わないようにすることである．現在でも，一部については転送情報や再送情報を用いて再送を削減しているが，さらなる削減が可能である．また，様々な部分マイグレーションの実装を行っていく必要がある．加えて，様々なビッグデータ処理を行う VM を用いて性能を測定することも計画している．

謝辞

本研究成果の一部は，国立研究開発法人情報通信研究機構の委託研究により得られたものです．

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, T. Kashiwagi, H. Kizu, and K. Kourai In Proceedings of the 2018 IEEE International Conference on Cloud Computing, pp.285-293 (2018).

[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 Dis-

tributed 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).