

OpenFlow による移動透過な 広域ライブマイグレーションシステムの提案と実装

藤本 大地^{†1} 近堂 徹^{†2} 前田 香織^{†1} 大石 恭弘^{†1} 相原 玲二^{†2}

概要: クラウドコンピューティング環境では、サーバ資源の効率的利用、動的な負荷分散、障害時の迅速な復旧といったものが求められている。その対策として仮想マシンのライブマイグレーション技術があるが、既存のライブマイグレーション技術は同一ネットワーク内での移動を前提としており、ネットワークを超える広域ライブマイグレーションを行うと通信途絶が発生する。現在、広域ライブマイグレーションを支援する技術として、MIPv6 を用いた kagemusha や ID/Locator 分離による LISP などが提案、実装されている。筆者らも、IP モビリティ技術のひとつである MAT を利用した広域ライブマイグレーション手法を提案しているが、これまで提案していた手法では、マイグレーションする仮想マシンや通信ノードの OS の改変を伴い、実用には高いハードルがあった。本研究では IP モビリティを OpenFlow により実装することで、ネットワーク側で広域ライブマイグレーションを支援する手法を提案し、システムの実装と評価を通じて、有用性を明らかにする。

キーワード: 広域ライブマイグレーション, OpenFlow, 移動透過通信, VM モビリティ

A Proposal and Implementation of a Wide-area Live Migration System with IP Mobility by OpenFlow

DAICHI FUJIMOTO^{†1} TOHRU KONDO^{†2} KAORI MAEDA^{†1}
YASUHIRO OHISHI^{†1} REIJI AIBARA^{†2}

Abstract: The cloud computing environments is needed things such as efficient use of server resources, dynamic load balancing, and rapid recoveries from failures. To achieve these, there is live migration technology of virtual machine as a countermeasure. However, communication disruption occurs when live migration beyond different networks over wide areas since most of the existing live migration technologies are assumed to move in the same segment networks. Currently, as a system to support wide area live migration, there are some approaches and implementation such as kagemusha with the MIPv6 and LISP with ID/Locator Separation. The authors also have proposed a wide-area live migration method using the MAT is one of the IP mobility technology, a method which has been proposed, with the modification of the OS in the virtual machine and the communication node to be migrated, and the practical use there was a high hurdle. In this paper, implemented by IP mobility using OpenFlow, we propose a method to support the wide-area live migration on the network side, through the evaluation and system and implementation, to clarify the usefulness.

1. はじめに

サーバの効率的な活用や障害時の可用性向上のために、仮想マシン (VM) のネットワークをまたがる広域ライブマイグレーションが有用であるが、通常ハイパーバイザで提供されるライブマイグレーションは同一ネットワーク内での移動を前提としているため、広域におけるライブマイグレーションを行うためには、ネットワークを移動しても通信途絶が発生しないようにする技術が必要となる。

現在、広域ライブマイグレーションを支援するシステムとして、Mobile IPv6[1] を継承した kagemusha[2] や ID/Locator 分離手法を採用した LISP[3]が提案され、実験環境での検証が進められている。筆者らも IP モビリティ技術のひとつである MAT(Mobility support Architecture and Technologies)を利用した広域ライブマイグレーション手法を提案している[4]。MAT の利点として、通信ノード間の最適経路通信をサポートしている点があるが、ホームアドレ

スとモバイルアドレスのアドレス変換を行うためにネットワークスタックに独自機構が必要となり、マイグレーションする VM の OS の改変を伴う。

そこで、本研究では OpenFlow[5]に着目し、移動透過通信機能を OpenFlow によってネットワーク側で実装する。これによって、VM の OS を改変することなく、ネットワークを越えて別の物理ホストへ移動させる「VM モビリティ支援機能」を提供することで、移動透過なマイグレーションを実現することを目指す。一方で、OpenFlow を用いて移動透過性を実現する際の VM のマイグレーション検知方法、通信途絶時間、アプリケーション継続への影響など明らかにしなければならない点もある。本研究では、プロトタイプシステムを実装することにより、マイグレーションによる通信途絶時間等の測定実験によってこれらの点を明らかにする。また、既存の広域マイグレーションの実装方法を比較し、開発した方法の有用性を示す。

本論文の構成は以下のとおりである。2 章では広域ライ

^{†1} 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University

^{†2} 広島大学情報メディア教育研究センター
Information Media Center, Hiroshima University

ブマイグレーションの既存研究を示し、広域ライブマイグレーションに対する要求要件をまとめる。3章では提案するVMモビリティ支援システムについて述べ、2章で定めた要求要件に対する解決策を示す。4章では3章のシステム構成に基づいたプロトタイプシステムの実装について示す。5章ではプロトタイプシステムを使った評価実験と考察について述べ、6章で本論文のまとめを行っている。

2. 広域ライブマイグレーションにおけるVMモビリティ支援

2.1 既存研究について

広域ライブマイグレーションによって発生する通信断への対策には大きく分けて3つの方法がある[6]。プロキシを経由し、VMの移動後のネットワークを移動元のネットワークに対してフォワードする「プロキシ」型、NAT機能下にVMを配置し、アドレス変換を行うことでVMと外部クライアントの通信を可能にする「NAT」型、移動元と移動先のネットワークをVPNによって同一セグメントとする「L2延伸」型である。

「プロキシ」型に分類される技術は、主に標準化技術であるMobile IP(MIP)[7]およびMobile IPv6(MIPv6)[1]を拡張して物理ノードからVMに適応させたもので、その代表としてHyperMIP[8]、kagemusha[2]などが挙げられる。MIPでは、プロキシ経由のトラフィックフォワードを行うことで、移動ノードのL3における接続性の維持、いわゆるIPモビリティを実現している。しかし、その技術特性から、一点障害、三角経路といった問題を潜在的に抱えており、MIPを拡張した技術もまた、その多くが、同様の問題点を引き継いでいて、先述の二例においても例外ではない。

「NAT」型に分類される技術として、MAT[9]、LISP[3]がある。MATは、移動ノードのカーネルに組み込まれ、L3における処理を拡張しNAT機能を組み込んでいる。拡張処理では主に、外向きに通信を行うためのモバイルアドレス(MoA: Mobile of Address)と、移動ノード上で動作するアプリケーションが恒久的に利用するためのホームアドレス(HoA: Home of Address)とを相互変換する。移動ノードがネットワークを移った際には新たなMoAを取得し、MoAとHoAの対応情報を更新することで、IPモビリティを実現している。このMATをVMに適用して広域マイグレーションする提案があるが[4]、VMごとにカーネル変更を行う必要があるため、実用的とはいえない。LISPはルーティングを効率化、簡素化するために開発されたルーティングプロトコルである。現在のインターネットにおけるIPルーティングは、IPアドレスという単一のアドレス空間によってデバイスのIDとネットワークの位置の2つを表現している。LISPではそれら2つの情報を分離し、デバイスのIDをEndpoint Identifier (EID)、ネットワークの位置をRouting Locator(RLOC)とすることで、RLOCを集約させ、ルーティ

ングの効率化を行っている。また、RLOCとEIDの分離によりノードの接続性がネットワークの位置に対して非依存となるため、IPモビリティが実現可能となる。一方で、該当ネットワーク全体のルーティングを入れ替えるため、機器の入れ替えが発生し、従来ネットワークに大幅な変更が必要になる可能性がある。これら「NAT」型では、アドレス空間を分離し、アプリケーションに使用するノードのアドレスを一意にすることで接続性を維持するが、マイグレーションを行うVMだけでなくそのVMの通信相手についても分離したアドレス空間の相互変換を行う必要がある。

「L2延伸」型は、基本的に広域イーサネットを構築する技術で、広域ライブマイグレーションにおける通信断を、同一ネットワーク間の移行であるL2ライブマイグレーションとすることで解消することができる。しかし、この型全般において、従来ネットワークに改変が伴うだけでなく、ブロードキャストドメインの拡大や通信経路の冗長化などの課題も生じる。

2.2 要求事項

前節で示した既存研究の問題点から、本研究で提案するVMモビリティ支援機能に対する要求事項を以下の3点に定めた。

- (1) トラフィックの集中に伴う通信一点障害、カプセル化による三角経路問題から解放される
- (2) VMを改変をしない
- (3) 既存のネットワークに大きな変更を加えず、VMモビリティ支援機能を提供する

まず、(1)については、プロキシ経由の通信によるトラフィックの一点集中、パケットのカプセル化による冗長経路に起因したVMに対する非効率な通信を解消するためである。(2)については、VMごとに改変を行う方式では実用性を損なうため、それを改善する必要がある。(3)については、システムの導入に対するハードルを下げ、より使いやすいものを提供するために必要である。

以上が、本提案における移動透過な広域ライブマイグレーションを行うための、システムの要求事項である。

3. VMモビリティ支援システム

3.1 システム概要

本論文にて提案するVMモビリティ支援システムは2.2の要求事項を満たした広域ライブマイグレーションを実現目的とするものである。提案システムでは、IPモビリティ機能を持つ仮想ゲートウェイを一般的な仮想化サーバ上に導入する。移動するVMがこの仮想ゲートウェイを介して通信を行うことにより、VMが別のネットワークにマイグレーションする場合においても通信を継続することが可能となる。2.2で示した要求事項に対して、本研究ではVMモビリティ支援システムにOpenFlow[5]を用いる。図1にOpenFlowを用いたシステムの構成概要を示す。図1では

仮想ゲートウェイが後述の OpenFlow スイッチ導入サーバに相当する。OpenFlow は、Software Defined Network (SDN) を実現するための技術の一つで、ネットワークを流れるパケットに対して、事前に設定しておいた条件に基づいて、パケットを転送、書き換え、破棄することが可能である。OpenFlow は、OpenFlow スイッチ (OFS) と OpenFlow コントローラ (OFC) の 2 つの要素で構成され、それぞれがパケット転送とフロー制御を担う。OFS は、ネットワークの各所に配置され、受信したパケットに対して OFC から与えられた条件をもとにパケット処理を行う。

近年では、関連研究[10]や[11]といった OFC の冗長化や分散化による単一障害点への対策や、SDN 対応製品による OFS のフロー制御のハードウェア化なども進められている。

3.2 OpenFlow を用いた類似研究

OpenFlow を用いて VM の広域マイグレーションを支援するものとして関連研究[12]がある。[12]は Portland や VL2 といった異なるネットワークアーキテクチャで構成された各データセンターのネットワークを抽象化し相互接続するために、OFC を制御する IaaS ミドルソフトウェアを提案している。これによってデータセンター間でのオペレーション、例えば VM の L3 マイグレーションを実行した際に、IaaS ミドルソフトウェアを介して各データセンターにある OpenFlow Controller へ制御命令を出すことで、OpenFlow で構築された経路を再構成し、VM の移動後も IP アドレスを変更することなく継続した通信を可能としている。

[12]のアプローチは、OpenFlow ネットワーク内で Hop-by-Hop ルーティングを行うことで VM の接続性を維持したマイグレーションを可能としている。一方で、筆者らの提案は複数組織が接続された IP ネットワークにおいて VM の IP 移動透過性を支援することを目的としており、同じ OpenFlow を使ったアプローチでも想定環境が異なる。

3.3 要求事項に対する解決策

提案システムが 2.2 で定めた要求事項に対してどのように解決するかについて述べる。(1)については、OpenFlow によるパケットの書き換え処理を使うことで、流れているパケットに対して直接宛先の変更を行うことができるため、カプセル化による冗長経路が発生せず、最適経路による通信が可能となる。また、ノード間の通信では、各ノードに

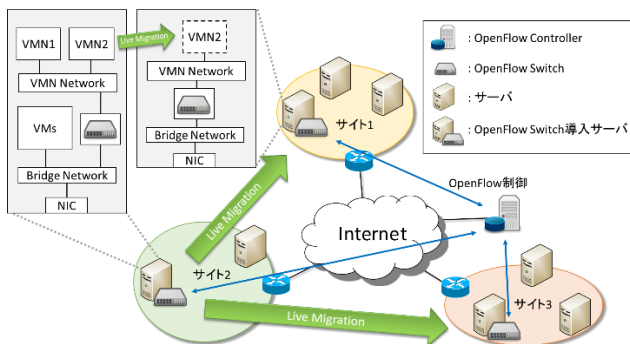


図 1: システム概要図

隣接した OFS を経由して通信を行うため、ある OFS に対して各所から集中的にトラフィックが転送されるということは発生しない。(2)については、OpenFlow は OFS と OFC から構成され、VM とは独立した要素からなるアーキテクチャであるため、VM に対して改変をする必要がない。(3)については、OFS を Edge Overlay 方式で導入することで解決を図る。Edge Overlay 方式であれば、Hop-by-Hop 方式と違い従来の基幹ネットワークを構成する機器に変更を加えることなく、ネットワークのエッジ部分、つまり、OFS を導入したサーバ内の仮想ネットワーク構成変更のみでよい。本論文では、ライブマイグレーションを行う VM を VMN (仮想移動ノード:Virtual Mobile Node) と呼称する。VMN に対して提供する移動透過マイグレーション支援機能は、MAT の MoA と HoA を参考にし、OpenFlow のパケット書き換え処理によってアドレスの相互変換を実現する。VMN に対して MoA と HoA という 2 つのアドレスを割り当てることから、本提案手法は 2.1 節で示した「NAT」型による手法といえる。そのため、通信を継続した広域ライブマイグレーションを行うには通信相手側にもアドレス変換を行うための仕組みが必要となる。したがって、本提案において VMN に対して継続した通信が可能なのは、OFS に接続された通信相手のみとなるが、アドレス変換機構は OFS で提供されるため、通信相手の VM にも一切の改変は必要としない。

以上より、接続性の維持が可能な通信相手に制約があるものの、本提案は 3 つの要求事項を満たしているといえる。

4. プロトタイプシステムの実装

4.1 概要

本プロトタイプシステムでは、オープンソースである OFC の Ryu と、OFS の Open vSwitch を用いて実装を行った。利用したバージョン等の実装環境を表 1 に示す。本システムにおける開発部分は、OFC で動作するメインプログラムであり、主な動作として、VMN の検出、VMN へのアドレスの割り当て、VMN 間通信の検出、VMN のライブマイグレーションの検出がある。次節以降で、動作の詳細を述べる。なお前提として、VMN となる VM にはあらかじめすべての VMN が同一サブネットとなるような静的なネットワークの設定が行われており、一意に定められたデフォルトゲートウェイと各 VMN で固有のプライベートアドレスが設定されていることとする。

4.2 管理情報

OFC で保持する管理情報をテーブル別に表 2 と表 3 に示す。表 2 の OFS 管理情報テーブルは OFS に付与された ID 毎に管理する。また、表 3 の VMN 管理情報テーブルは VMN のプライマリインターフェースの MAC アドレス毎に管理する。どちらのテーブルも、OFS と VMN の数に応じて生成される。OFS 管理情報テーブルは、VMN に割り当てる

表 1：実装環境

要素	Software および Version
OpenFlow Switch	Open vSwitch 2.3.0
OpenFlow Controller	Ryu 3.14
使用言語	Python
OpenFlow Protocol	version 1.3.0
Hypervisor	QEMU/KVM

表 2：OFS 管理情報テーブル

フィールド	説明
アドレスプール	利用可能なアドレスのプール
ゲートウェイ	ネットワークの WAN 側ゲートウェイの MAC アドレス
データパス情報	OFS (データパス) の基本情報

表 3：VMN 管理情報テーブル

フィールド名	説明
モバイルアドレス	VMN に割り当てられているモバイルアドレス
ホームアドレス	VMN が持っているホームアドレス
データパス ID	現在 VMN のいるネットワークにある OFS の ID
通信相手リスト	現在 VMN が通信している通信相手の MAC アドレスのリスト

ためのアドレスプール、VMN のルーティングを正しく行うためのデフォルトゲートウェイの MAC アドレス、アドレス変換フローの書き込みなどに用いるためのデータパス情報で構成されている。VMN 管理情報テーブルは、VMN に恒久的に付与される HoA、動作するネットワーク毎に割り当てられている MoA、VMN が現在接続している OFS の ID であるデータパス ID、そして VMN の通信相手の MAC アドレスが格納された通信相手リストで構成されている。

VMN 管理情報テーブルは、VMN のプライマリインターフェースの MAC アドレスをキーとしているため、通信相手リストの MAC アドレスから通信相手の VMN の管理情報を参照できる。

4.3 基本動作

プロトタイプ実装における VMN の検出および MoA の割り当て動作フローを図 2 に示す。VMN の検出は、VMN が送信する ARP request パケットを、OFS で受信することから始まる。OFS は受信した ARP request パケットを OFC へ送り (パケットイン)、メインプログラムによりパケットの解析を行う。パケットからは、VMN に割り当てられているプライベートアドレス (HoA) や MAC アドレスのほか

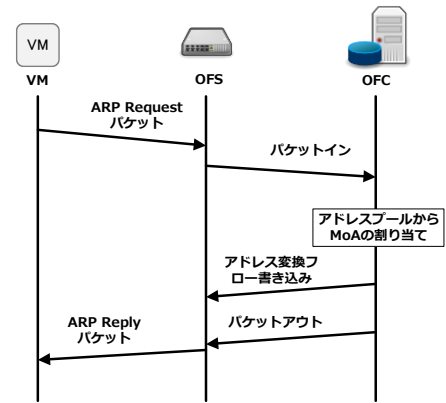


図 2：VMN の検出および MoA の割り当て動作フロー

に、送信元 OFS の情報が得られるので、OFC は VMN の検出時にその VMN がどの OFS 配下に接続されたものかを把握することができる。パケットの解析後、OFS 管理情報テーブルのアドレスプールから、送信元 OFS で利用可能なアドレスの一つを VMN に MoA として割り当て、送信元 OFS に対して VMN の MoA と HoA を相互変換するためのアドレス変換フローを書き込む。ARP request パケットに対する応答は、OFS 管理情報テーブルから得られるゲートウェイ MAC アドレスを格納した ARP reply パケットを生成した後、OFS へ送信 (パケットアウト) し、VMN へ転送される。

提案システムでは、VMN 間通信の検出については、ある VMN が異なるネットワーク上の VMN と通信をする際に、単純な ARP 解決を行うことができないという問題がある。L2 延伸技術の場合、すべてのパケットをカプセル化しているため、ARP 解決が可能であるが、OpenFlow の場合は、パケットの転送を従来の IP ルーティングで行うため、ARP のようなブロードキャストパケットはセグメントを超えることができない。そこで、OFS が他の VMN に対する ARP 解決を検出した際に、VMN 管理情報テーブルから通信相手となる VMN の MAC アドレスを格納した ARP reply パケットを生成し、パケットアウトすることで ARP 解決を行っている。

VMN のライブマイグレーション時の動作フローを図 3 に示す。ライブマイグレーションが発生し、VMN がレジュームする際、Hypervisor(移動先)は、アドレス重複の確認や ARP テーブルの更新をかけるために、Hypervisor 上で VMN の RARP request パケットを代理送信する。本システムでは、この Hypervisor から送信される RARP request パケットを VMN のマイグレーションの検出に用いる。RARP request パケットをパケットインすることで、OFC は移動した VMN の MAC アドレスが得られる。OFC では VMN 管理情報テーブルから、該当する MAC アドレスの VMN が移動前に接続していた OFS と、通信相手の VMN を求める。その後、移動前の OFS と移動前の通信相手が接続していた OFS に

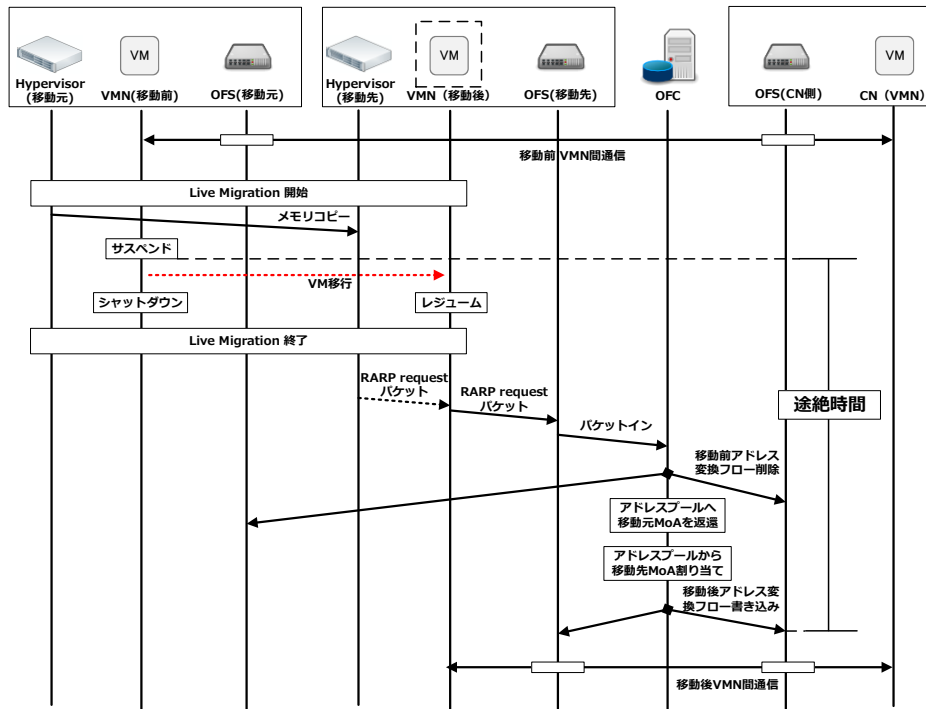


図 3 : VMN のライブマイグレーション時動作フロー

対して、移動した VMN の移動前アドレス変換フローを削除する。そして、移動後の VMN が属する OFS に対して、移動前の通信を維持させるように、MoA の割り当てを行った後で、VMN に新しく割り当てた MoA に対応したアドレス変換フローの設定を自動で行う。これにより VMN が移動しても、通信を維持することが可能となる。

プロトタイプシステムにおけるアドレス変換と VMN 間での通信イメージを図 4 に示す。VMN1 で通信に利用する IP アドレスは送信元アドレス(src addr), 宛先アドレス(dst addr)ともに双方の HoA が入っている。VMN1 が接続している OFS にてアドレス変換が行われると、この src addr と dst addr フィールドを HoA から MoA へ直接書き換え、通信相手のネットワークへと IP ルーティングにより転送さ

れる。通信相手のネットワークまでパケットが転送されると、VMN2 が接続している OFS で MoA から HoA への変換が行われ、VMN2 が受信する。逆方向についても同様の処理が行われ、互いに通信をすることが可能となる。

5. 評価実験

5.1 評価実験の目的

評価実験では、最初に本システムの通信回復までのオーバヘッドを調べるために、プロトタイプシステムにより、移動透過マイグレーション支援機能を提供している VMN が L3 ライブマイグレーションを行った際に、システムによって通信相手 (CN : Correspondent Node) との通信が回復されるまでの途絶時間 (ダウンタイム) を測定し、同様の環境における L2 ライブマイグレーションとの比較を行う。次に、得られたデータと既存研究との比較から、提案システムの有効性を評価する。

5.2 ローカル環境実験

最初に OFC とサブネットワークを切り分けている Router, VM を稼働させるホストが同一サブネットワークにあるローカル環境で実験を行った。実験構成図を図 5 に示す。実験では、ホスト 1 の上で動作する VMN01 からホスト 3 上で動作する VMN02 に対してトラフィックを発生させ、VMN01 がホスト 1 からホスト 2 へマイグレーションを行う。提案システム (L3 マイグレーション) における途絶時間と、通信回復までのオーバヘッドを求めるため、ホスト 1-ホスト 2 間で同一ネットワークの L2 マイグレーション (L2 マイグレーション) を行った場合の 2 パターンについて測定する。L2 マイグレーションの際は、各サーバの一つ目の上位のブリッ

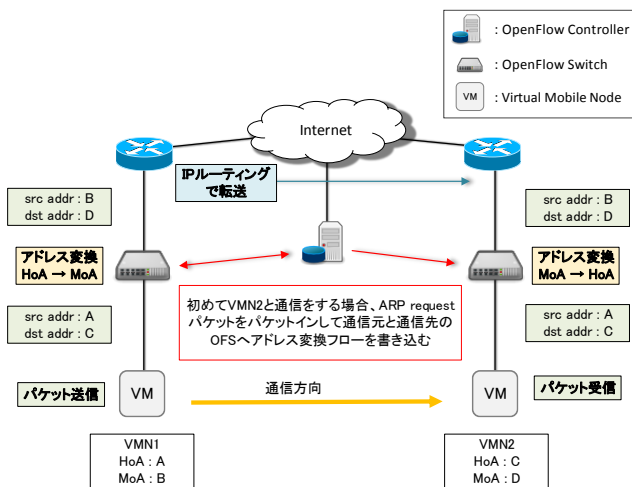


図 4 : OpenFlow を用いたアドレス変換による VMN 間通信

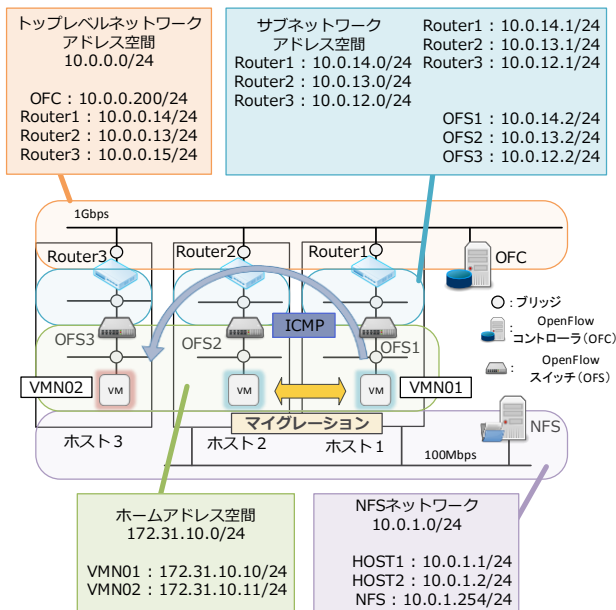


図 5：ローカル実験環境

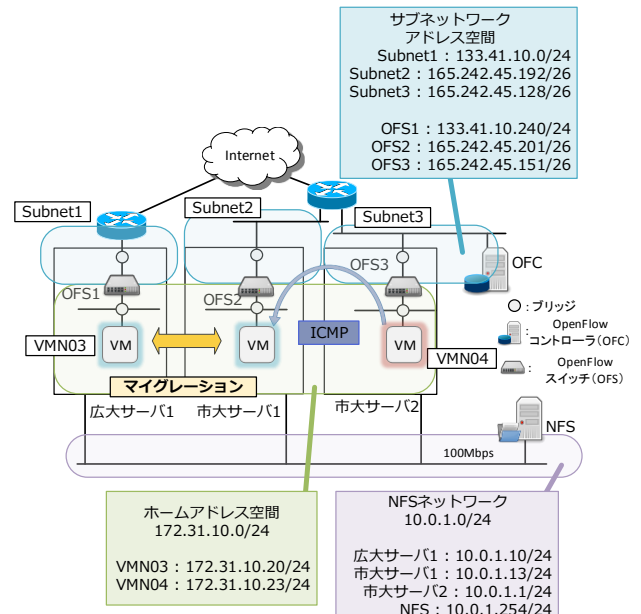


図 6：広域実験環境

ジに VMN を接続し、L3 ライブマイグレーションでは、VMN 間は上位ルータをまたがるマイグレーションを行った。各 OFS は図 5 中の OFC と接続を確立しており、VMN の検出を終え、VMN01 と VMN02 は通信可能な状態にある。

測定方法は次の通りである。VMN01 から VMN02 に対して、測定パケットを送信しながら、Host1 と Host2 の間をライブマイグレーションさせる。測定では 10ms 間隔で送信する ICMP を利用し、途絶時間は連続して欠落した ICMP パケット数と送信間隔から算出する。また、VMN がライブマイグレーションする際に、Hypervisor 側で VM を一時的に停止させる時間が存在する。本論文での実装環境では Hypervisor として KVM を使っており、KVM の VM 一時停止時間は初期値が 30ms となっている。この一時停止時間が VM の途絶時間に影響を及ぼす可能性があるため、この値を設定可能な最小値である 1ms に設定を変更した。また、マイグレーションする VM には RAM 1GiB、ストレージ 10GiB 程度のとても小さなものを用意した。

5.3 広域環境実験

次に、VMN が異なる L3 ネットワークをまたいでマイグレーションする広域環境での実験を行った。広域における実験環境を図 6 に示す。測定の目的は、インターネット経由における提案システムの動作検証とローカル環境との性能比較を行うためである。測定における事前設定や前提、測定方法はローカル環境と同様であるため、割愛する。

5.4 測定結果

各測定環境における途絶時間の測定結果を表 4 に示す。表 4 のデータは、CN 側の VMN から ping を実行し、得られたパケット送信回数とパケット受信回数の差およびシーケンスログから連続した欠落 ICMP パケットを得て、各測

表 4：測定結果

測定内容		途絶時間[ms]
ローカル環境	L2 マイグレーション	146
	L3 マイグレーション	173
広域環境		209

定の 10 回分の平均値をとったものである。ローカル環境における L2 マイグレーションと提案システムの測定結果から、提案システムの途絶時間が 27ms 分増加していることが確認できた。この時間は主に OFS から OFC に対して RARP request パケットをパケットインする時間と、OFC から OFS に対してアドレス変換フローが書き込まれるまでにかかる時間であると考えられる。

次に広域環境の測定結果から、提案システムの場合、ローカル環境での L2 マイグレーションに対して、サブネットをまたがるマイグレーション時のオーバーヘッドが 63ms であることが確認できた。

5.5 考察

次に、表 5 に方式ごとのマイグレーション時の通信途絶時間の結果を示す。表 5 中のオーバーヘッドは各種システムを用いずに測定した場合とシステムを導入した場合との途絶時間の差分と定義し、VMN-CN 間 RTT は測定環境の規模の指標として掲載している。まず、既存研究の測定方法についてまとめる。HyperMIP[8]は、ローカル環境にて L2 マイグレーションと L3 マイグレーションの測定を行い、測定ツールとして apache bench と tbench を使うことで、出力されるベンチマーク結果から途絶時間が算出されている。kagemusha[2], LISP[13], MAT[4]はいずれも提案システムの測定と同様に連続して欠落した ICMP パケット数と送信間隔から途絶時間が算出されている。VXLAN[14]では、

表 5 : 途絶時間およびオーバーヘッド比較表

アプローチ(測定内容)		測定方法	途絶時間 [ms]	オーバーヘッド [ms]	VMN-CN 間 RTT[ms]
提案システム	ローカル環境 L2 マイグレーション	ping 10ms	146	—	3.241
	ローカル環境 L3 マイグレーション		173	27	
	広域環境		209	63	1.675 (移動前) 12.41 (移動後)
HyperMIP [8]		ab, tbench	3230	480	—
Kagemusha [2]		ping6 100ms	3000	2000	0.300
LISP [13]		ping 20ms	320	50	16.47 (移動前) 27.27 (移動後)
MAT [4]		ping6 10ms	2540	1571	28.50 (移動前) 31.70 (移動後)
VXLAN [14]	ローカル環境	UDP 10ms (測定プログラム)	478	—	0.860
	広域環境		502		30.00

※ LISP のデータ : 文献の途絶時間を短くするように改良を加えた[13]のデータを参照

ping(ICMP)の代わりに UDP を使って測定を行っており、データ部に格納したシーケンス番号を利用して欠落した UDP パケット数を導出し、送信間隔と掛け合わせることで途絶時間が算出されている。

提案システムのオーバーヘッドは、ローカル環境の L3 マイグレーションの途絶時間から L2 マイグレーションの値を差し引いた 27ms といえる。一方で、広域ライブマイグレーションの場合、インターネット経由で動作することでおおよそ 40ms 程度のオーバーヘッドが発生している。マイグレーションを検知する OFS と OFC の間の RTT などが直接影響するため、サイト間の遅延が大きくなるとオーバーヘッドはさらに増加すると考えられる。

提案手法と既存研究との比較について述べる。表 5 の結果から提案システムは、途絶時間は最も短く、オーバーヘッドでは各手法の中でも十分に小さい結果となった。だが、途絶時間は各手法の計測環境や設定パラメータに大きく依存するため、比較評価の指標として適当ではない。よって、オーバーヘッドについての比較が妥当である。提案システムのオーバーヘッドが小さい要因としては、事前に OFS にて利用可能なアドレスを予約することにより、VMN に対して即座に新たな MoA を割り当てが可能となり、アドレス割当てに要する時間を短縮できたことが挙げられる。事前のアドレス予約については、広域におけるライブマイグレーションの有効な利用シーンを考慮すると、何らかの方法で移動先ネットワークやホストで使用可能なリソースを把握する際に可能であると考えている。既存研究においては、HyperMIP は VM のトラフィックを Hypervisor へ経由させ

るための処理負荷がオーバーヘッドになっている、kagemusha ではアドレス重複確認に要する時間がオーバーヘッドに入っている。この時間(約 1 秒) [2]を差し引いても提案システムの方が小さい。MAT は VM に付与された MoA の更新にかかるオーバーヘッドが大きいため、提案システムより大きい結果となっている。また、LISP の結果は、提案システムと同程度である。

6. おわりに

本論文では、移動透過な広域ライブマイグレーションを実現するために、OpenFlow を用いた VM に対する移動透過機能を提案し、プロトタイプシステムを用いた評価から、その性能を示した。本提案手法では、既存ネットワークに対して OpenFlow 機能が動作可能な仮想スイッチをアドオンする形を採用している。これにより、大幅なネットワーク変更を必要とせず、かつマイグレーション対象 VM の OS を改変することなく広域ライブマイグレーションを実現することが可能となった。また、提案システムにおいて、VM のマイグレーションを検出する手法を明らかにし、実装を行った。

評価実験では、OpenFlow を用いることによる影響について、通信途絶時間の観点から評価を行った。測定の結果、提案システムのオーバーヘッドは既存研究と同等かそれ以上の結果を出した。さらに、インターネットを経由する広域環境においても、オーバーヘッドの著しい増加は発生せず、アプリケーションセッションが継続できることを確認した。また、MAT における VM の OS 改変や、LISP や VXLAN に

よるルーティングの入れ替えに伴うネットワーク改変を回避することで要求事項を満たす実装を達成した。

今後の課題としては、複数 VM 環境における性能評価やアプリケーションの通信継続への影響、本提案手法によるライブマイグレーションを前提とした VM 管理方法の検討などが挙げられる。

謝辞

本研究の一部は日本学術振興会科学研究費助成金 15K00130 の支援を受けて実施しています。

参考文献

- 1) D.Johnson et al, "Mobility Support in IPv6," Internet Engineering Task Force (IETF), RFC3775, June 2004.
- 2) 広瀬崇宏他, "仮想マシンに対して透過的な Client Mobile IPv6 トンネリング機構," 電子情報通信学会論文誌 B, J95-B(10), pp.1239-1252, Oct 2012.
- 3) D.Farinacci et al, "The Locator/ID Separation Protocol (LISP)," Internet Engineering Task Force (IETF), RFC 6830, Jan. 2013.
- 4) Tohru Kondo, Toshiaki, et al. "A Mobility Management System for the Global Live Migration of Virtual Machine across Multiple Sites." Proc. of IEEE Computer Software and Applications Conference Workshops 2014, pp.73-77, July 2014.
- 5) OpenFlow Switch Version 1.3.0, June 2012.
<http://www.opennetworking.org/images/stories/downloads/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf>
- 6) Fischer, Andreas, et al. "Wide-area virtual machine migration as resilience mechanism." Reliable Distributed Systems Workshops (SRDSW), 2011 30th IEEE Symposium on. IEEE, Oct 2011.
- 7) C. Perkins, et al, "IP Mobility Support for IPv4", RFC3344, Aug 2002.
- 8) Li, Qin, et al. "HyperMIP: Hypervisor controlled mobile ip for virtual machine live migration across networks." IEEE 11th High Assurance Systems Engineering Symposium, pp.80-88, Dec 2008.
- 9) 相原玲二他, "アドレス変換方式による移動透過インターネットアーキテクチャ," 情報処理学会論文誌, Vol.43, No.12, pp.3889-3897, Dec. 2002.
- 10) 黒木圭介, 林通秋, "OpenFlow Controller における冗長化についての考察," 電子情報通信学会ソサイエティ大会講演論文集 2012 年 通信(2), No.115, Aug, 2012.
- 11) 梁轟, 川島龍太, 松尾啓志. "複数 OpenFlow コントローラにおける動的負荷分散手法の検討," 電子情報通信学会総合大会講演論文集 2014 年 通信(2), No.163, Mar, 2014.
- 12) Boughzala, Bochra, et al. "OpenFlow supporting inter-domain virtual machine migration." Wireless and Optical Communications Networks (WOCN), pp.1-7, May, 2011.
- 13) Raad, Patrick, et al. "Achieving sub-second downtimes in internet-wide virtual machine live migrations in LISP networks." Integrated Network Management (IM 2013), Proc. of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp.286-293, May 2013.
- 14) MOHAMMAD, TAHA, and CHANDRA SEKHAR EATI. "A Performance Study of VM Live Migration over the WAN." Electrical Engineering, pp.70, May 2015.