

OpenFlow による移動透過通信支援システムの提案

藤本 大地[†] 大石 恭弘[‡] 林 直樹[‡] 前田 香織[‡] 近堂 徹[§] 相原玲二[§]

[†]広島市立大学情報科学部 [‡]広島市立大学大学院情報科学研究科

[§]広島大学情報メディア教育研究センター

概要: サーバの効率的な活用や障害時の可用性向上のために、ネットワークをまたがった仮想マシンの広域ライブマイグレーションが有用であるが、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。そのために仮想マシンに移動透過通信アーキテクチャを使うアプローチがあるが、本研究では OpenFlow を用いて移動透過通信を支援するシステムを提案する。本システムは移動透過通信のメリットを生かしつつ、OpenFlow で収集できる情報も活用した効率的な広域ライブマイグレーションの基本機構として動作することを目的としている。

A Proposal of an IP Mobility Support System Using OpenFlow Framework

Daichi Fujimoto[†] Yasuhiro Ohishi[‡] Naoki Hayashi[‡] Kaori Maeda[‡]
Tohru Kondo[§] Reiji Aibara[§]

[†]Faculty of Information Sciences, Hiroshima City University

[‡]Graduate School of Information Sciences, Hiroshima City University

[§]Information Media Center, Hiroshima University

1. はじめに

近年、Microsoft Azure, Amazon EC2 といったクラウドサービスが注目されているが、クラウドコンピューティング環境は、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では同一ネットワーク内での移動を前提としているため、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してしまう。

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、MIPv6 (Mobile IPv6) を継承した Kagemusha[1]や ID/Locator 分離手法を採用した LISP[2]などが提案され、実環境での検証・導入が進められている。

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[3]。MAT の利点として、最適経路通信をサポートしている点が挙げられるが、エンドホスト間で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、通信を行う端末や仮想マシン間で OS の改変が伴う。

そこで、本研究では OpenFlow[4]に着目し、移動透過通信アーキテクチャを OpenFlow により実装する。これにより、マイグレーションする仮想マシンや通信端末に対して移動透過通信機構を導入することなく、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ためのシステムを提案する。

2. 移動透過通信アーキテクチャ MAT

MAT は移動端末(MN)に移動透過機能を提供する IP モビリティ技術のひとつである。MAT はアドレス変換テーブルを持つ IMS (IP address Mapping Server)と MN に組み込まれるアドレス変換モジュールからなる。MN に MoA (モバイルアドレス)と HoA (ホームアドレス)の2つのアドレスを割り当てることで、MN の位置及び ID の識別を行う。アドレスの組は IMS に保管され、MN が移動した場合、新たなアドレスの組を IMS へ通知する。その後、通信相手(CN: Correspondent Node)にアドレスの更新を通知し、CN が IMS を参照することで MN 移動後も継続して通信が可能となる。本来 MAT は物理的な移動端末をターゲットに移動透過性を提供してきたが、仮想化技術の発達により仮想マシンの広域マイグレーションへの適用が現実的なものとなりつつある。しかしながら、これを実現するためには仮想マシンの改変が必要となり、広く展開するのは難しい。

3. 提案システムの概要

提案システムでは、ネットワーク側で移動透過機能を実現するために、仮想マシンのマイグレーションをトリガとして OpenFlow のパケット書き換え処理を利用する。これにより、既存の仮想マシンを改変することなく所望の目的を達成することができる。

図 1 にシステムのネットワーク構成例を示す。移動ノード MN (Migration Node) はマイグレーションをする仮想マシンで、HOST1 と HOST2 の間をマイグレ

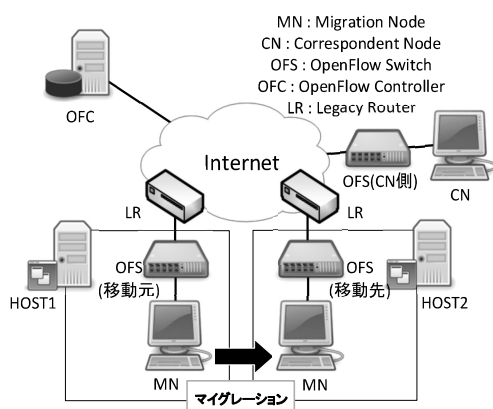


図 1: ネットワーク構成例

ーションする。HOST1と HOST2 は異なるネットワークに位置し、MN と CN はともに OFS (OpenFlow Switch)を介した通信が可能なネットワークを想定している。

各 OFS は一つの OFC (OpenFlow Controller)とチャネルを確立しており、初期設定を終え、フローを持っている状態を初期状態とする。OFC では、フロー単位で MAT によるアドレス変換処理を実現するためのアルゴリズムを実装する。なお、提案手法では IP ヘッダの書き換えを前提とし、各 OFS の間に従来ルータ LR (Legacy Router)がいる場合でも動作することを目標とする。

4. 動作フロー概要

図 2 に、図 1 の環境で MN がライブマイグレーションする際のシステムの動作フローを示す。(1) MN の移動先にいる OFS は、MN が移動してきた際に送信する RARP パケットを受信し、OFP (OpenFlow Protocol)の Packet-In メッセージを使って OFC に RARP パケットを送信する。(2) OFC は、その RARP パケットの MAC アドレスをもとに、MN に対して送信する ARP パケットを作成する。作成した ARP-Request を OFP の Packet-Out メッセージを用いて送信することで、ARP-Reply から MN の HoA を取得する。(3)OFS は ARP-Reply を OFC に送信、それを OFC が解析し、MN の新たな MoA の取得を行うことで、移動先の新たな MoA と MN 固有の HoA を対応させる。その後、(4)OFC は、OFP の flow-mod メッセージを使って、各 OFS が持つシステム用のフローを新たな MoA に書き換えるように更新メッセージを送信する。図 2 では移動先と CN 側の 2 つの OFS に対して MoA 更新通知を送信しているが、他にも OFS があれば、すべての OFS に対して MoA 更新通知が送信される。OFC は flow-mod メッセージを送信した後、各 OFS が変更を適用したことを知るために Barrier メッセージを OFS に送る。(5)OFC が各

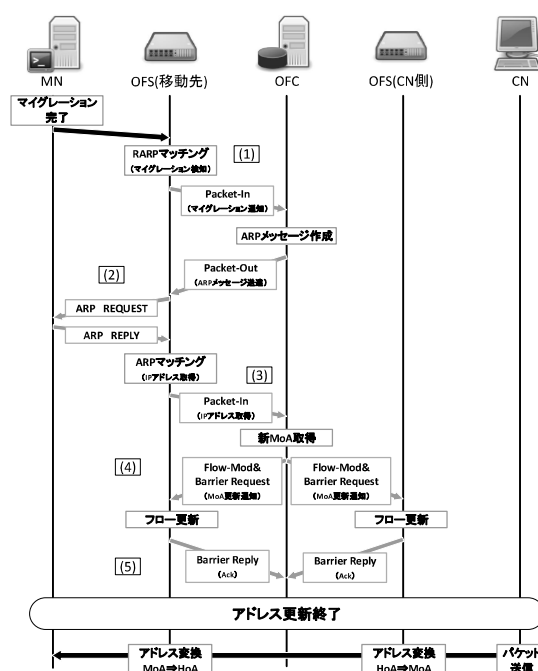


図 2: 動作フロー図

OFS からの Barrier-Reply を受け取ることでアドレス更新が終了する。

5. まとめと今後の予定

本稿では OpenFlow を用いた仮想マシンの広域ライブマイグレーション支援システムの提案を行った。今後はシステムの開発、評価を行っていく。

評価では、既存手法[3]である移動端末に導入した場合との途絶時間の比較を行う。加えて、OpenFlow により発生するオーバーヘッドとして、OFS で保持するフロー数に対するスケーラビリティや OFC の処理時間等についても評価を行う。

さらに、OpenFlow の利点であるフロー単位での情報把握や制御を視野に入れ、よりアプリケーション特性を考慮した広域ライブマイグレーションの実現を検討する。

参考文献

- [1] 広瀬他, “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, vol. J95-B, no. 10, pp. 1239-1252, Oct. 2012.
- [2] D. Farinacci, et al. “The Locator/ID Separation Protocol (LISP),” IETF, RFC 6830, Jan. 2013.
- [3] T. Kondo et al. “A Mobility Management System for the Global Live Migration of Virtual Machine across Multiple Sites,” Proc. of 2014 IEEE 38th Computer Software and Applications Conference (COMPSAC), pp. 73-77, Jul. 2014
- [4] OpenFlow, Open Networking Foundation, <https://www.opennetworking.org/> (参照 2014-10-09).

謝辞

本研究の一部は日本学術振興会科学研究費助成金 24300027, 24500083 の支援を受けて実施しています。