Vol.2017-OS-140 No.1
2017/5/16

マルチテナントシステムの開発を支援するIaaS環境

古橋 健斗[a)]　松本 拓也　福田 浩章[b)]

概要：クラウドサービスでは，アプリケーションを提供するサービス提供者が物理マシンやネットワークを保有するインフラ提供者から必要に応じてリソース (e.g., 仮想マシン) を確保し，サービスを提供している．サービス提供者は，最大負荷 (必要になる仮想マシンの最大数) を見積もることでサービスの円滑な運用を目指しているが，予め見積もることは難しい．一方，インフラ提供者は，物理マシン，仮想マシンの負荷状況 (e.g, CPU やメモリ使用量) をもとに仮想マシンを再配置し，データセンタ全体の運用効率向上を目指している [1][2]．この検証には実運用に適用することが望ましいが，サービス提供者の SLA を保証する必要があり，実現は難しい．また，大規模なデータセンタを準備することも難しいため，シミュレーションでの検証を行わざるをえない [3]．そこで本研究では，RaspberryPI を利用し，サービス提供者，インフラ提供者それぞれの要求を容易にテストできる環境を提供する．具体的には，複数の RaspberryPI を使用した仮想データセンタの構築と負荷状況の監視，仮想マシンの操作を実現する．また，必要に応じて仮想マシンを増減し，スケールアウトする機能を実現する．そして，仮想マシンの移動，スケールアウトを本環境で実行し，その機能性能を示す．

IaaS environment to support multi-tenant system development

Kento Furuhashi[a)]　Takuya Matsumoto　Hiroaki Fukuda[b)]

1. はじめに

仮想化技術の進化を背景に，物理マシンを保有してサービスを展開する形態 (シングルテナント) から，必要に応じてリソース (e.g., CPU やメモリ) を確保しサービスを展開する形態 (マルチテナント) に移行している．そして，これらは Amazon Web Service (AWS) や Microsoft　Azure のようなクラウドサービスを利用して提供されている．マルチテナント環境では，図 1 に示すようにアプリケーションを提供するサービス提供者が物理マシンやネットワークを保有するインフラ提供者から必要に応じてリソース (e.g., 仮想マシン) を確保し，サービスを提供している．サービス提供者は，最大負荷 (必要になる仮想マシンの最大数) を見積もることでサービスの円滑な運用を試みるが，予め正確に見積もることは難しい．そのため，実際には経験に基づき最大数を設定し，運用している．一方，インフラ提供者は，物理マシン，仮想マシンの負荷状況 (e.g, CPU やメモリ使用量) をもとに仮想マシンを再配置し，データセンタ全体の運用効率向上を目指している [1][2]．この検証には実運用に適用することが望ましいが，サービス提供者の SLA を保証する必要があり，実現は難しい．また，大規模なデータセンタを準備することも難しいため，シミュレーションでの検証を行わざるをえない [3].

一方，近年小型計算機の開発を背景に，それらを複数台利用してクラスタとして利用する事例や，SDN の検証に用いる例が見受けられる．小型計算機は一般の計算機やサーバと比較して安価で，複数台用意することも十分に可能である．また，仮想化機能を備えた CPU を搭載する小型計算機も存在する．

そこで本研究では，小型計算機の一つである RaspberryPI(RasPI) を利用してデータセンタを模擬し，インフラ提供者が効率の良いデータセンターの運用方式 (e.g., アルゴリズム) を開発するテスト環境を提供する．また，予め最大数を見積もることなく，物理マシン，仮想マシンの状況に応じて動的に仮想マシンを複製し，動的に負荷を分散する機構 (オンデマンドスケールアウト) を実現する．そ

a) ma17099@shibaura-it.ac.jp
b) hiroaki@shibaura-it.ac.jp

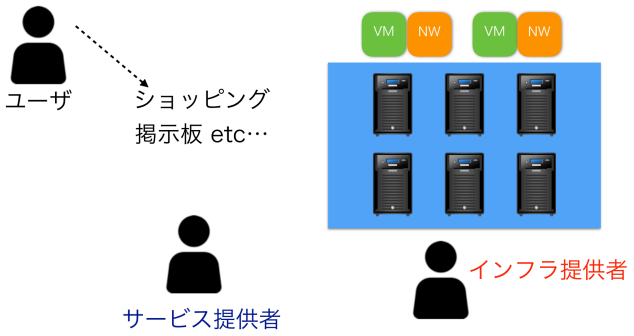

図 1　マルチテナント環境

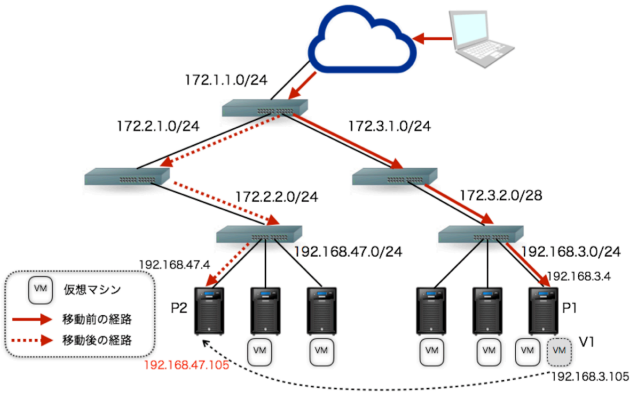

図 2　想定するクラウド環境

して，負荷分散や運用効率を判定する指標となる CPU やネットワークの負荷の取得する機能，および仮想マシンの複製や移動を実行する機能を API として提供する．最後に，これらの API を使用して仮想マシンの移動，スケールアウトを実行し，テスト環境の基本性能を示す．

以降，2 節では本研究で想定するクラウド環境について概説し，テスト環境の要件を整理する．3 節では本研究のアプローチについて概説し，4 節で実装について述べる．次に 5 節ではテスト環境での実験と結果をまとめる．そして，6 節で関連研究について述べ，最後 7 節でまとめと今後の課題を述べる．

2. クラウド環境とテスト環境の要件

本節では，本研究で想定するクラウド環境を概説し，それらの実現に必要なテスト環境の要件を述べる．

2.1 クラウド環境

本研究で想定するクラウド環境を図 2 に示す．クラウド環境では，一般に複数のネットワークスイッチとそれらに接続した物理マシンで構成される．そして，サービス提供者からリソース取得の要求があると，インフラ提供者は物理マシンで仮想マシンを起動してそれらを提供している．クライアントのリクエストはスイッチで適切に転送され，対象となる仮想マシンが適切に処理してサービスを運用している．ここで，物理マシンが高負荷になり，仮想マシンを移動する状況を想定する (図 2)．図 2 では，物理マシン P1 で仮想マシン V1 が動作している状況において，V1 を物理マシン P2 に移動することを想定している．P1 と V1 にはそれぞれ IP アドレス (192.168.3.4, 192.168.3.105) が設定されており，ユーザの立場からは区別できない．一方，P2 にも 192.168.47.4 の IP アドレスが設定されている．仮想マシンの移動では，一般に IP アドレスは変化しないため，V1 の IP アドレスは P2 に移動した後も変わらず 192.168.3.105 となる．V1 の移動は P2 が直接接続しているネットワークスイッチ，および上流に位置するスイッチに影響しないため，従来 V1 に届けられていたパケットは依然として 192.168.3.0/24 を管理するスイッチに届けられることになり，移動後の V1 に届けられることはない．移動後の V1 に正しくパケットを届けるためには，V1 の IP アドレスと移動後のネットワークに合致したものに変更 (e.g., 192.168.47.105) する必要がある．加えて，ユーザからはこの変更は透過であるため，V1 へのリクエストは依然として 192.168.3.105 宛に送信される．しがって，ネットワークスイッチに変更を加え，宛先 192.168.3.105 へのパケットは変更後のアドレスに書き換えて (192.168.47.105) 送信する必要がある．

このように，クラウド環境で円滑にサービスを運用するためには，仮想マシンの移動や複製に伴い，関連するネットワークスイッチも連動して変更する必要がある．

2.2 テスト環境の要件

2.1 節で述べたように，本研究で想定するクラウド環境の実現には，仮想マシンの移動だけでなく，IP アドレスの変更など，移動に伴う変更や，移動の契機となる状況の把握，多数のマシンを制御できる必要がある．本研究では，クラウド環境を実現するため，以下の項目をクラウド環境の必要要件と考える．

負荷状況の把握　仮想マシンの移動や複製は，一般に高負荷状態にある物理マシンの負荷を分散し，仮想マシンで提供するサービスの性能維持や，低負荷状態の物理マシンに散財する仮想マシンを集約し，リソースの効率的な使用ののために行われる．本テスト環境でもこれらの負荷状態を測定するため，負荷状態の判定に一般的に用いられる CPU とメモリ使用率を使用する．また，物理マシンの負荷分散を目的として仮想マシンを移動させる場合，移動先物理マシンを決定する必要があるが，ネットワークを利用したサービスに主に用いられるクラウド環境ではネットワーク使用量も移動先を決定する指標となり得る．そこで本研究では，物理マシン単体のネットワーク使用量だけでなく，スイッチ単位でのネットワーク使用量を測定し，移動先決定の指標に用いる．

表 1　RaspberryPi2 ModelB の性能

CPU	ARM Cortex-A7 クアッドコア 900MHz
メモリ	1GB
ネットワーク	10/100Mbps イーサネット
電源	900mA(4.5 ~5.5W)

IP アドレスとルーティング管理　仮想マシンが移動する時，移動先物理マシンが同一ネットワークに所属している場合には IP アドレスの変更は必要ない．しかし，2.1 節でも述べたように，異なるネットワークに所属する物理マシンに移動する場合には移動後に IP アドレスの変更が必要になる．同一ネットワーク内だけの移動/複製は，クラウド環境の効率的な運用の妨げになるため，本研究ではネットワーク間を跨いだ仮想マシンの移動/複製も想定する．その場合，複数の物理/仮想マシンに同一の IP アドレスを指定することはできないため，仮想マシン移動時には IP アドレスを開放し，移動後には未使用の IP アドレスを設定できる必要がある．さらに，ネットワーク間を跨いだ仮想マシンの移動/複製には，2.1 節で述べたように，IP アドレスの変更に伴い，関連するネットワークスイッチを変更し，適切にクライアントからのリクエストを転送する必要がある．

複数マシンの管理　本研究では，クラウド環境の効率的な運用のため，仮想マシンの移動/複製を効率的に行うアルゴリズム開発/検証にテスト環境を利用することを想定している．そのため，本テスト環境は多数の物理マシン，ネットワークスイッチで構成することになり，それぞれの状態管理 (e.g, 負荷の測定) や制御 (e.g, 設定変更や仮想マシンの移動/複製) を個別に行うことは現実的には難しい．そこで，テスト環境を構成する物理マシンやネットワークスイッチを一元管理し，設定変更や制御を用意に実現できる必要がある．

3. テスト環境の実現

本節では，RasPI で仮想環境を実現する方法について述べた後，2.2 節で挙げた要件に対応する本研究での実現方法を述べる．

3.1 RaspberryPI での仮想化環境

本研究のテスト環境では 1 に示す RaspberryPI2 ModelB を利用し，Linux 系専用 OS である Raspbian Jessie Lite を実行する．CPU である ARM Cortex-A7 は仮想化拡張機能を備えているため，Linux カーネルが備えるハイパーバイザ，KVM を実行することができる．一般の PC では，KVM はパッケージ管理ソフトウェア (e.g., apt-get) などを利用してインストールできるが，RasPI ではパッケージでは提供されていない．そのため，カーネルの組み込み

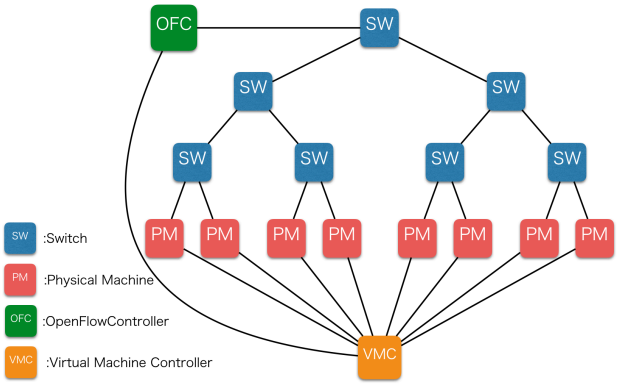

図 3　テスト環境のアーキテクチャ

機能として実行する．また，KVM での仮想マシンの管理には virt[4] を利用することが一般的であるが，RasPI では virt を利用することができない．そのため，本研究では KVM と連携して動作する qemu が提供するコマンドを利用する．

一方，2.2 節でも述べたように，本テスト環境の実現には仮想マシンの移動/複製に同期したネットワークの変更が必要になる．近年，仮想化技術の進歩に伴い，ネットワークを柔軟に制御するために Software Defined Network (SDN)[5] が注目を集めており，利用されている．そこで本研究でも SDN を利用してネットワークを制御する．本研究では SDN の一種であり，広く利用されている OpenFlow[6] を利用するため，OpenFlow 対応のソフトウェアスイッチである Open vSwitch(OvS) を RasPI にインストールして利用する．そのため，本テスト環境はすべて RasPI だけで構成できる．

3.2 テスト環境のアーキテクチャ

2.2 節で述べたように，多数の物理マシンや仮想マシンの状態管理，および制御を個別に行うことは困難である．そこで，本研究では，仮想マシンや仮想マシンを起動する物理マシンを一元管理する仮想マシンコントローラ (VMC) を実現する．また，KVM を用いた仮想マシンの移動には NFS を利用する必要があり，一般に NFS の利用はローカルネットワークであることが望ましい．そこで本テスト環境では，図 3 に示すように，ユーザがサービス利用のために利用するネットワークとは異なる管理ネットワークを設け，すべての物理マシンを管理ネットワークにも接続する．そして，この管理ネットワークに VMC を接続することで，後述する負荷状況や IP アドレス，およびルーティングの管理を容易にする．

3.3 負荷状況の把握

負荷状況の指標として，各物理マシンの CPU 使用率とメモリ使用率，各スイッチのトラフィック量を取得する．物

表 2　API で使用できる機能

仮想マシンの起動	http://IPaddress:8000/vmStart?id
複製	http://IPaddress:8000/clone?from?to
移動	http://IPaddress:8000/migrate?from?to
リソースの取得	http://IPaddress:8000/getResource?id

理マシンの CPU およびメモリ使用量は物理マシンの/proc ディレクトリに格納されているファイルからリソース使用量を計算するプログラムを実装し，一定間隔 (デフォルトは 10 秒毎) で実行してその値をデータベースに格納する．また，トラフィック量の取得には，各 OvS が計測している統計情報を元に各 OvS 全体，ポート毎のトラフィック量をそれぞれ取得する．そして，これらの情報を Open Flow Controller である OFC に送信し，OFC のデータベースに格納する．VMC は OFC のデータベースにアクセスを行うことで，トラフィック量を取得することができる．

3.4　IP アドレスとルーティング管理

テスト環境では，使用中の IP アドレスを VMC で動作するデータベースで管理し，仮想マシンの起動や複製，移動時に次の手順で決定し，実際に設定する．まず，移動先物理マシンが所属するネットワークアドレスを調べ，使用中の IP アドレス一覧を取得する．次に，ネットワークアドレス，ブロードキャストアドレス，使用中の IP アドレスを除きランダムで決定しする．最後に，起動，複製，移動後の仮想マシンの ifconfig，route コマンドを使用して IP アドレスとデフォルトルートを設定する．

次に，異なるネットワークに仮想マシンの移動や複製を行う場合，OpenFlow を利用し，経由する OvS の設定を変更することで，パケットの転送経路を変更する．具体的には，仮想マシンを移動する時，OFC はすべての OvS のルーティング情報を保持ししているため，移動前アドレスのルーティングテーブルが存在する OvS を探索し，パケットを転送するポートを変更すると同時にパケットの送信先アドレスを変更する．そして，移動後の仮想マシンから戻るパケットに関しては，送信元アドレスを移動前のアドレスに書き換えて送信する．また，仮想マシンを複製する時には，移動と同様の処理を 1/2 の割合で実行することにより，ラウンドロビンで負荷を分散する．

3.5　複数マシンの管理

3.2 で述べた通り，VMC を使用することで，多数の物理マシンや仮想マシンの状態管理や制御を個別に行うことが可能になる．本テスト環境の利用者は，VMC で立ち上げた Web サーバにアクセスするだけで，各物理マシンの操作が可能になる．また，アルゴリズムのテストを行うために，表 2 に示す負荷状況の把握や仮想マシンの操作が行える WebAPI を実装した．仮想マシンの起動では，指定した id の物理マシンに仮想マシンを起動する．仮想マシンの複製と移動では，from に指定した id の仮想マシンを to に指定した id の物理マシンに複製・移動を行う．リソースの取得では，id に指定した物理マシン，仮想マシンのリソースを json 形式で取得する．この WebAPI を使用することで，仮想マシンの起動，移動，複製と各物理マシンのリソースの取得を行うことが可能になる．

表 3　サーバを通信する時の経由する OvS の数

route	通信間	経由する OvS 数
1	Server1 ⟷ Server2	1
2	Internet ⟷ Server1	3
3	Server1 ⟷ Server3	3
4	Server1 ⟷ Server8	5

表 4　仮想マシンの移動の合計時間内訳

内訳	時間
仮想マシンの移動の開始から完了	約 33 秒
ネットワーク構成変更	約 3 秒
遅延時間	約 2 秒

4.　評価

本節では，本テスト環境の妥当性についての評価を行う．今回，実験は図 3 のような環境で行う．

仮想マシンの複製や移動が発生した場合のルーティングが行われるまでの転送時間を iperf[7] を用いて計測した．最大 TCP セグメントサイズを 128Byte/256Byte/512Byte/1024Byte/1400Byte の 5 段階に設定し，表 3 に示す状況においてそれぞれ計測した．仮想マシンの移動を実際に行うと，仮想マシンの移動の発生から完了，その後にネットワーク構成変更まで平均して合計約 38 秒かかった．仮想マシンの移動の実験には，図 3 の環境で 1 つのスイッチを経由するルートで行なった．作成した API を用いて仮想マシンの移動を行うと，表 4 のような結果になった．仮想マシンの移動の実行は単体テストの際に約 33 秒かかる事が判明している．仮想マシンの移動の開始から完了するまでは，移動元の仮想マシンが動作中であるため，ネットワークが切れることはなく，仮想マシンの移動の完了後，移動元の仮想マシンが停止し，ネットワークの構成変更時に初めてネットワークが遮断される．実験の結果，ネットワークの構成変更には約 3 秒かかる事が結果として得られている．その他にかかる約 2 秒は VMC から OFC へ，移動が発生したと通知する時にかかる通信による遅延時間だと考えられる．結果，仮想マシンの移動の発生からネットワークの構成変更までに合計約 38 秒の時間がかかるが，ping が通じなくなり，ネットワークが遮断される時間は約 4 秒であった．仮想マシンの複製実行時には，複製が実行されてから，VM1 へ IP アド

表 5　仮想マシンの複製の合計時間内訳

内訳	時間
仮想マシンの移動の開始から完了	約 41 秒
ネットワーク構成変更	約 4 秒
遅延時間	約 2 秒

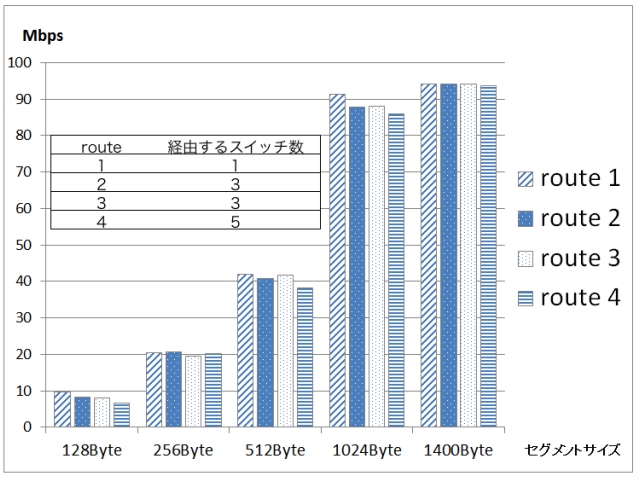

図 4　ネットワーク速度

レスを変化しながらアクセスを繰り返し，VM2 へ初めてリクエストが送信されるまでには約 47 秒の時間がかかった．仮想マシンの複製の実験においても，図 3 の環境で 1 つのスイッチを経由するルートで行なった．作成した API を用いて仮想マシンの複製を行うと，表 5 のような結果になった．仮想マシンの複製においても，複製に約 41 秒ほど，ネットワークの構成の変更に約 4 秒，仮想マシンの移動の場合と同原因と思われる遅延時間が約 2 秒発生した．仮想マシンの移動の場合とは異なり，仮想マシンの複製では複製後も，複製元仮想マシンが動作しているため，ネットワークが遮断されることがなかった．

ネットワークの速度は図 4 の示す結果となった．

route1 と route4 を比較すると分かるように，経由する OvS の数によって僅かに転送速度が低下する．また，図 4 で示す，セグメントサイズが 1024Byte と 1400Byte にした際のトラフィック速度に大きな変化は見られなかった．

トラフィック量の取得についても iperf を用いて検証を行い，iperf でネットワークに負荷を加え，本研究で実装したトラフィック量取得の機能を用いて接続される OvS 全てのトラフィック量を取得し，続いて，ネットワーク全体のトラフィック量を取得を行った．これら値は，

OvS1+OvS2+OvS3+...+OvSx = ネットワーク全体のトラフィック量 = iperf で転送したトラフィック量

となる事が明白であり，本環境において，上記した式の結果を得ることができたため，正確な値であると言える．

セグメントサイズが 1024Byte と 1400Byte にした際のトラフィック速度に大きな変化は見られなかったが，RasPI のイーサネットは 10/100Mbps ポートであるため 100Mbps 以上の転送速度を得ることはできない．したがって正確な値であると測定でき，仮に 1Gbps の転送速度性能を持つスイッチであれば 10 倍の転送速度を期待でき，仮想マシンの移動や複製にかかる時間も低下する．仮想マシンの移動や複製におけるネットワークの構成変更にかかる処理時間は，RasPI より高性能な物理マシンでは高速になる．実際に 1Gbps ポート，Intel Core 2 Duo である CPU を搭載した高性能物理マシンにおいて，仮想マシンの移動と複製を行うと表 6 の結果となった．表 6 に示すように，仮想マシンの移動では本環境のおよそ 15 分の 1 の転送速度，複製ではおよそ 12 分の 1 の転送速度であった．この差はネットワーク速度，また処理性能の差により発生するものであり，本環境を高性能物理マシンで実行すると同等の実行時間を得ることができる．

表 6　高性能物理マシンでの実行時間

	移動	複製
実行時間	2.23 秒	6.01 秒

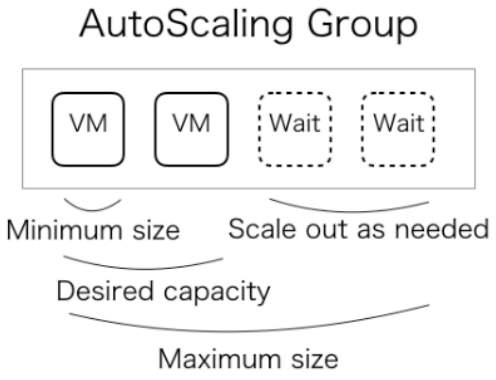

図 5　AutoScaling

また，route1 と route4 のように経由する OvS の数によって転送速度が低下する問題についても本環境に限らず生じる現象であり，本研究において実装した OFC によって発生する低下は非常に少ないと考えられ，ボトルネックになることは無いと言える．

5. 関連研究

本研究の関連研究として，AWS や OpenStack などのクラウド環境を構築できる IaaS が挙げられる．しかし，これらのサービスでは，本研究の目的であるインフラ提供者とサービス提供者の要求を満たすことができない．

5.1 AWS[8]

IaaS のクラウドコンピューティングサービスである AWS では，仮想マシンと同様のインスタンスをサービス提供者にリソースとして提供している．AWS には様々なサービスが提供されており，Amazon Elastic Compute Cloud(Amazon

EC2) というインスタンスを提供するサービスや，Amazon Simple Storage Service(Amazon S3) というクラウドストレージを提供するサービスが存在する．サービスの中には，AutoScaling という負荷がかかった場合に EC2 インスタンスを自動で増減させ，スケールアウト/スケールインを行う機能や，Elastic Load Balancing というロードバランスを行う機能が提供されている．AutoScaling は，図 5 に示すように，AutoScaling Group を作成し，EC2 インスタンスの最小数と最大数を設定し，処理負荷がかかった場合に，待機状態の EC2 インスタンスを立ち上げ，ロードバランスを行い，負荷分散を行なっている．また，処理負荷が少なくなった場合には，EC2 インスタンスを待機状態にし，リソースを有効活用できるようにしている．しかし，この AutoScaling Group は異なるネットワークを跨いで作成することはできない．したがって，ロードバランスを行う仮想マシンのネットワークは同一である必要があり，ネットワーク内のすべての EC2 インスタンスが過負荷状態になってしまった場合には対応できない．また，事前に設定した EC2 インスタンスの最大数を超える処理負荷がかかってしまった場合，全ての EC2 インスタンスが正常に起動しなくなってしまうことがある．

加えて，AWS は EC2 インスタンスを起動している物理マシンのリソース情報はブラックボックスになっており，実際に計測することができない．また，物理マシン上でどのように EC2 インスタンスが動作しているか，トラフィック量の取得が不可能など，テスト環境としては不明瞭な点が多い．したがって，AWS でテストを行うと，実際の環境で動作した時の予測をすることができないといった問題点が存在する．

5.2 OpenStack[9]

OpenStack はオープンソースで開発されているクラウド環境構築用のソフトウェア群である．OpenStack では，仮想マシンとストレージ，ネットワークといったリソースを提供するクラウド環境を構築することが可能である．OpenStack は様々なコンポーネントの組み合わせで成り立っている．中でも，仮想マシンを管理する Nova や仮想ネットワークを提供する Neutron が本研究と関連している．OpenStack では仮想マシンを起動，停止や移動を行うことができるが，仮想マシンの複製はリンククローンしか使えず，ネットワークと同期して仮想マシンの複製を行うことはできない．また，性能を上げる方法として，物理マシンの性能を上げるスケールアップは行うことができるか，スケールアウトは行うことができない．したがって，サービス提供者の自身のサービスの最大負荷見積もりという要求を満たすことができない．また，特定の物理マシンのリソースの取得や特定のスイッチのトラフィック量を取得することができないといった問題があるため，アルゴリズムのテストを行うには不十分であると言える．

6. まとめ

クラウド環境で利用される仮想マシン再配置に関するアルゴリズムの開発は，データセンタなどを保持しない組織や個人では非常に困難であり，アルゴリズムを開発しても，実験や検証が困難な状況であった．そこで，本研究では安価で小型な RasPI を用いることで，省スペースで低コストなデータセンタの構成を提案，構築した．本環境では仮想マシンの移動や複製の仮想化技術を容易に利用可能な環境の構築を行なった．結果，低コストでデータセンタの構成を実現した．

6.1 今後の課題

今後の課題を以下にまとめる．本環境は全て CUI で動作するため，一目して，本環境の挙動が理解しにくい．そこで，物理マシン，物理マシンで動作する仮想マシンを含めたネットワークのトポロジをグラフィカルに表示し，ネットワークのトラフィック量もグラフを用いて表示可能な GUI の作成．また本環境では DHCP は実装しておらず，RasPI の IP アドレスはユーザが指定する必要があるため，DHCP 機能の追加．以上の機能の拡張が考えられる．また，現在は VMC，OFC 間の通信は現在 ssh を用いて行っており，コントローラ間の通信が全体の実行速度低下の原因となる．そのため，コントローラ間の通信を行う独自のプロトコルの作成も今後の課題として挙げられる．

参考文献

[1] Takahiro Hirofuchi, Hidemoto Nakada, Satoshi Itoh, Satoshi Sekiguchi, "Reactive Cloud: Consolidating Virtual Machines with Postcopy Live Migration", IPSJ Transactions on Advanced Computing Systems, Vol.5 No.2 8698 ,2012
[2] 広渕崇宏, 中田秀基, 伊藤智, 関口智嗣, "高速マイグレーションを利用した仮想マシン配置最適化システムの検討", 情報処理学会研究報告 IPSJ SIG Technical Report, Vol.2010-OS-115 No.14, 2010/8/4
[3] H. Kim, J. Kim and Y. B. Ko, "Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking," 16th International Conference on Advanced Communication Technology, Pyeongchang, 2014, pp. 758-761. doi: 10.1109/ICACT.2014.6779064
[4] virt-manager website. https://virt-manager.org/ 2017/04/19
[5] Open Netrowking Foundation website. https://www.opennetworking.org/ 2017/01/11
[6] Open Netrowking Foundation OpenFlow website. https://www.opennetworking.org/sdn-resources/openflow 2017-01-11
[7] iperf website. https://iperf.fr/
[8] Amazon Web Services website. https://aws.amazon.com/jp/ 2017/04/19
[9] OpenStack Documentation. https://docs.openstack.org/ 2017/04/19