

# 緊急災害情報に基づく OpenFlow を用いた バックアップシステムの実装と評価

原 瑠理子<sup>†</sup> 小口 正人<sup>†</sup>

<sup>†</sup>お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

E-mail: †ruriko@ogl.is.ocha.ac.jp, ††oguchi@is.ocha.ac.jp

あらまし 近年、クラウドコンピューティングが普及し、データセンタ事業者が提供するパブリッククラウドと、自社内に構築するプライベートクラウドを組み合わせたハイブリッドクラウドが注目されている。プライベートクラウドは安全性が高く、パブリッククラウドは拡張性が高いことからそれぞれのクラウドの利点を使い分けることで、1つの環境で効率的に作業することができる。しかし、大規模な自然災害などが発生する場合、膨大なデータがビッグデータ処理基盤に流れ込み、アクセスも集中することで、短時間に大きな負荷がかかる。そのため、クラウド内・クラウド間における迅速な切り替えの対応が重要となってくる。そこで本研究では、緊急地震速報をもとにバースト的な負荷変動を予測し、実際に地震が発生した時刻からシステムに負荷が掛かるまでの短い間に投機的な制御を行うことで、緊急災害時に発生する問題の解決を図る。

キーワード SDN, OpenFlow, OpenStack, クラウド, トラフィックエンジニアリング

## Implementation and Evaluation of Backup System using OpenFlow Based on Emergency Disaster Information

Ruriko HARA<sup>†</sup> and Masato OGUCHI<sup>†</sup>

<sup>†</sup> Ochanomizu University 2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610, JAPAN

E-mail: †ruriko@ogl.is.ocha.ac.jp, ††oguchi@is.ocha.ac.jp

**Key words** Software-Defined Network, OpenFlow, OpenStack, Cloud, Traffic Engineering

### 1. はじめに

近年、インターネットやセンサ技術の普及、また携帯型デバイスの発展に伴い、大量のデータを処理するクラウドやデータセンタも増加し、ビッグデータへの対応が社会における情報処理基盤において重要となってきた。ビッグデータ処理では、実世界を反映する形で短時間に非常に大きなシステム負荷の変動が起こり得る。例えば、大規模な地震が発生した緊急災害時には、センサから膨大なデータが情報処理基盤に流れ込み、安否確認や避難経路の情報を収集したり発信したりするユーザによってアクセスが集中する。実際に、2011年3月11日に発生した東日本大震災では、情報を得ようとする県民や県外の人たち、また世界中からのアクセスが集中したことで、福島県のホームページのサーバが処理できずにダウンする事態が発生した。この時、震災前の平均アクセス数の約10倍ものアクセス数が記録されている。

現在、データセンタ事業者が提供するパブリッククラウドと、自社内に構築するプライベートクラウドを組み合わせたハイ

ブリッドクラウドが注目されているが、このような短時間に大きな負荷がかかる場合、手動による迅速な切り替えは難しい。すなわち、情報インフラとして重要な役割を果たしているシステムを、緊急災害時にも安全に稼働するために、クラウド内・クラウド間における迅速な切り替えや、重要なデータをバックアップするといった対応が重要となってくる。このように、情報インフラとして重要な役割を果たしているビッグデータ処理基盤は、緊急災害時にも停止することなく稼働し、情報提供することが期待される。

そこで本研究では、このようなバースト的な負荷変動を、緊急地震速報[1]を始めとする外部情報から予測し、OpenFlowコントローラを用いて、どのようにネットワークトラフィックを制御すべきか判断する手法を検討する。

### 2. クラウド環境の普及とその現状

#### 2.1 ハイブリッドクラウド

近年、クラウドコンピューティングの1つの形態として、ハイブリッドクラウドが普及している。ハイブリッドクラウドと

はプロバイダなどが提供するパブリッククラウドと、企業などが自社内で利用するために構築したプライベートクラウドを組み合わせて、効率的に利用することができるクラウドである。例えば、Amazon の AWS で知られるようなパブリッククラウドは、キャンペーンサイトといった一時的に大量の処理を必要とする際には、拡張性に優れているため適している。また、プライベートクラウドは企業内でサーバを管理するクローズドなシステムとなるため安全性が高く、コントロールを維持できる。したがって、この2つのクラウドを併用することで、それぞれの特性を活かしたパブリッククラウドを構築することができる。

## 2.2 ハイブリッドクラウド導入の課題

しかしながら、ハイブリッドクラウドを導入している企業は実社会においてまだ少ない。この原因の1つとして、複雑化するシステムに対する手動による制御の限界が挙げられる。現在、ハイブリッドクラウドの運用管理コストを最小化するための検討 [2] が進んでいるが、迅速な対処が要する場合には対応できていない。特に、大規模な自然災害が発生した際には、急激なシステムへの負荷が起り得る。この場合、拡張性のあるパブリッククラウドに迅速に切り替えたり、重要度の高いデータを扱うプライベートクラウド内でレプリケーションを行うなど、短時間の間にシステムをコントロールすることでシステムへのダメージを抑えることができる可能性がある。緊急地震速報が発令された場合、数十秒ないし数分後に大規模な地震が到達する事が予測される。その短い時間の間に、重要なデータをバックアップノードに複製するなどシステムを防御する対策を取ることができれば、極めて有用である。特に地理的に分散したハイブリッドクラウド環境においては、ダメージが予想される地点のクラウドから、安全と思われるクラウドへ、可能な限りデータをコピーすれば、被害を最小限に抑えられる可能性が考えられる。しかし、このような迅速な制御や大規模で複雑化したクラウド環境の制御を、手作業で行うのは難しい。

そこで本研究では、Twitter から発信される緊急地震速報の情報をトリガとし、投機的にクラウド内・クラウド間で自動化した制御を行うことで、自然災害などの不測の事態へ対応する (図 1)。

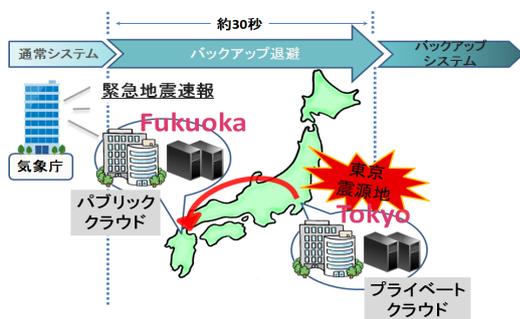


図1 本研究の提案システム概要

## 3. ハイブリッドクラウド環境の構築

### 3.1 OpenStack

本研究では、IaaS(Infrastructure as a Service) と呼ばれるクラウドサービスを提供する、クラウド環境構築用のオープンソースソフトウェアである OpenStack [3] を用い、実験用のクラウド環境を構築した。ハイブリッドクラウドの構築に OpenStack を採用した理由は3つある。第一に、パブリッククラウド・プライベートクラウドどちらとも基盤として構成可能であったこと。第二に、複数のコンポーネントの組み合わせで作られているので、環境に応じて必要な機能のみを柔軟に利用可能であること。第三に、多くの企業が導入しているため活発な議論や開発が行われていること。

本研究の実験環境は Controller ノード、Network ノード、Compute ノードの3つのノードで構成されており、それぞれのノードに導入したコンポーネントがうまく連携することで、仮想環境を構築することができる。本研究では、Controller ノード1台、Network ノード1台、Compute ノード4台の計6台のサーバを用いて仮想環境を構築した (図 2)。このクラウド間を繋ぐことで、プライベートクラウドとパブリッククラウドを接続したハイブリッドクラウド環境を模擬している。また、本研究の実験環境ではノードを分割しているが、全てのコンポーネントを1台のサーバの上に集約した構成や、サーバの台数に応じた構成が可能である。Controller ノードは OpenStack 環境全体を管理するコントローラとしての役割を、Network ノードは外部ネットワークとインスタンスの間のネットワークを制御するネットワークサービス、Compute ノードは仮想マシンインスタンスを実行させるためのメモリやストレージを提供する。

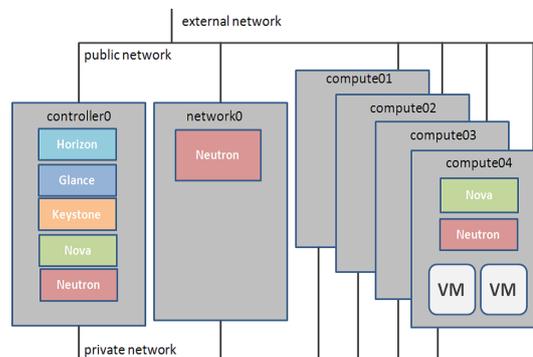


図2 OpenStack によるクラウド構築

本研究の実験に用いた各ノードの性能を表 1 に示す。OpenStack の Controller ノード、Network ノード、Compute ノードの全てに同一のマシンを用いた。

表1 マシン性能

OS	Linux3.13.0-43-generic
CPU	Intel(R) Xeon(R) CPU E3-1270 V2 @ 3.50GHz 4C/8T
Memory	16GB
HDD	500GB

また OpenStack は様々なコンポーネントが連携することで、クラウドサービスの機能を実現している。本研究では OpenStack のバージョン IceHouse を用い、そのバージョンに対応するコンポーネントの内、表 2 に示す 5 つのコンポーネントを導入したクラウド環境を構築した。このコンポーネントが連携する全体像を図 3 に示す。

表 2 OpenStack コンポーネント一覧 (IceHouse)

コンポーネント	機能
Nova	仮想サーバの管理
Glance	ゲスト OS の管理
Keystone	統合認証
Horizon	Web 管理コンソール
Neutron	仮想ネットワーク管理

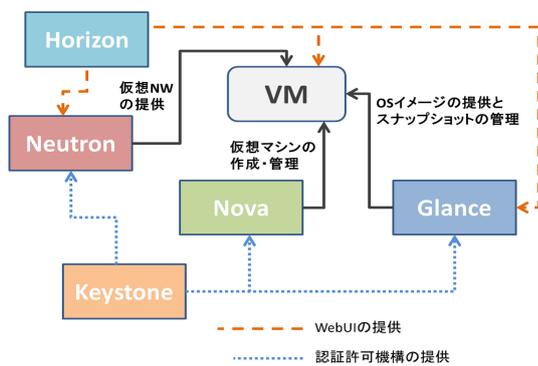


図 3 各コンポーネントの働き

#### 4. 緊急地震速報

緊急地震速報 [1] とは、気象庁が中心となって提供している、地震発生後に大きな揺れが到達する数秒から数十秒前に警告を発することができる地震早期警告システムである。震源に近い地震計が最初の揺れ (P 波) を感知すると、気象庁に情報を送り、瞬時に予想される震源や地震の規模を計算して速報として知らせる。P 波の方が S 波よりも早く伝わるため、大きな被害をもたらすことが多い遅れて伝わってくる S 波を、先に伝わってきた P 波を利用して知らせることが可能となる。したがって、緊急地震速報の知らせを受けてから大きな揺れ (S 波) が到達するまでに、30 秒程度の猶予時間がある。この数十秒の間に、システムへの何らかの災害に対する対処を行うことは有用であると考えられる 4。

本実験では、この緊急地震速報 bot を模擬した Twitter アカウントを作成し、同じような形式で地震情報を流し、テストを行った。この発信されるツイートの取得にあたっては、Twitter 社が提供している「Twitter Developers」を利用した。これは、Twitter の機能を使ったサービスを開発したいユーザ達のために、Twitter API を提供する。本実験はこの API を利用して OAuth 認証による、緊急地震速報の情報取得を可能とした。

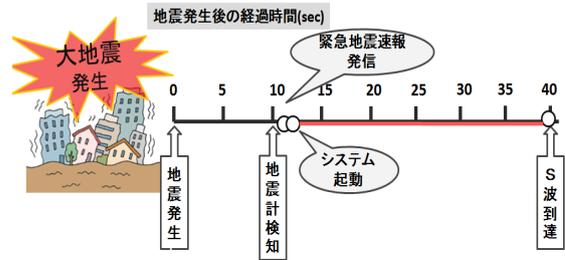


図 4 地震発生時の流れ

#### 5. 緊急災害時制御モデルの提案

本実験での制御モデルを提案する (図 5)。本実験は自社内に構築したプライベートクラウドが東京にある場合を想定する。また、パブリッククラウドは東京から遠く離れた遠隔地に置くものとする。震度と震源地により、以下のように異なる対応を行う制御モデルである。

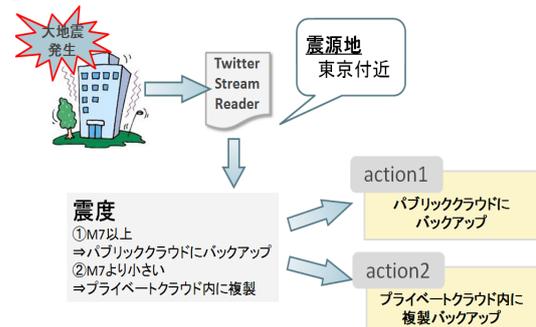


図 5 制御モデル

- (1) パブリッククラウドにバックアップ (M7 以上)
- (2) プライベートクラウド内に複製バックアップ (M7 未満)

(1) パブリッククラウドにバックアップ  
震源地が東京付近かつマグニチュード 7 以上の大きさの地震を観測した場合、遠隔地にサーバを置くパブリッククラウドにバックアップを行う。

阪神淡路大震災で M7.3、東日本大震災で M9.0 の大きさの地震を観測したことから、M7 以上の大地震の際には建物自体が損壊し、サーバが物理的に直接ダメージを受けることが想定される。そこで、緊急地震速報を検知し実際に大きな揺れが到達する数十秒の間に、より多くのデータを遠隔地にあるパブリッククラウドにバックアップ処理することを目標とする。

(2) プライベートクラウド内に複製バックアップ  
震源地が東京付近かつマグニチュード 7 未満の大きさの地震を観測した場合、自社内のプライベートクラウド内にバックアップを行う。

物理的に直接ダメージを受けないとしても、アクセスが通常時よりも集中することが想定されるので、安全性は確保しつつ自社内の別のサーバにバックアップを行う。

#### 6. SDN/OpenFlow

本研究のバックグラウンドトラフィック制御に用いた OpenFlow 技術 [4] について説明する。SDN(Software-Defined Network) とはネットワークの構成、機能、性能などをソフトウェアの操作

だけで動的に設定, 変更できるネットワーク, あるいはそのためのコンセプトを指す [5][6]. SDN を用いると, 物理的に接続されたネットワーク上で, 別途仮想的なネットワークを構築することが容易に実現可能になる. 仮想的なネットワークを構築することで, ネットワークの物理的な制約から離れて, 目的に応じたネットワークを柔軟に構築しやすくなる. そのような環境においては, トラフィックの変動に応じて動的にネットワークの構成を変更するといった, プログラマブルな制御が可能になる.

### 6.1 OpenFlow コントローラ

OpenFlow では図 6 のように, ネットワーク全体の経路制御をコントローラと呼ばれる機器上のソフトウェアで集中管理し, スイッチではデータ転送機能のみを実行する. 物理ネットワーク・仮想ネットワークの両方をコントローラで集中管理することによって, 既存のネットワークで実施していた各スイッチでの経路制御の設定が不要となり, ネットワークの単純化と運用および管理の負荷の大幅な削減を実現する. また, コントローラによるネットワークの集中管理により, 物理ネットワーク・仮想ネットワーク構成の動的な最適化が可能となる.

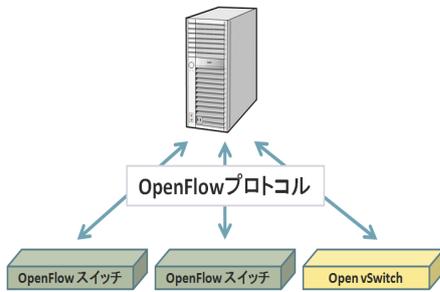


図 6 OpenFlow コントローラによる制御

### 6.2 Ryu コントローラフレームワーク

Ryu コントローラフレームワーク (以下, Ryu) は, SDN アプリケーションの開発に必要なライブラリやツールを提供するフレームワークである [7]. データプレーンを制御するための基本機能や, SDN アプリケーションで共通的に必要となる機能を提供することで, 開発をより容易にする. Ryu は, SDN アプリケーションの開発に必要なライブラリやツールを提供する OpenFlow コントローラフレームワークである. 他の OpenFlow コントローラに比べて様々なプロトコルに対応している. また, Ryu は元々は集中型のコントローラであったが, 大規模なシステムではボトルネックになってしまう懸念から, バージョンアップに伴って分散型のコントローラに変更された.

### 6.3 OpenFlow を用いた先行研究

先行研究では, ネットワークのトラフィック量の変動に応じてネットワークの構成や帯域をプログラマブルに制御する検討が進んでいる [8]. また, 災害時に優先度に基づいて動的に回線を選択し, その切り替えを OpenFlow を用いて行うもの [9] や災害発生後に OpenFlow が通信状況の異常を検知しリンクを切り替えるもの [10] がある. しかし, これはどれも異常発生後に OpenFlow による制御を行う. これらの技術は一般的に緩やかな負荷変動に対して行うことを想定しているため, 短時間に起

こる大きな負荷変動に耐えることは難しい. しかし実際には, 一旦重い負荷が生じてからシステム再構成を速やかに行うことは難しく, 障害が起きた後やトラフィック量の変動が起きてからでは遅い. そこで本研究では緊急地震速報をトリガとし, ハイブリッドクラウド上のインスタンスに接続している Open vSwitch(以下, 同様) を Ryu コントローラで自動的に制御し, 投機的に迅速なバックアップを行う.

## 7. クラウド環境での OpenFlow 制御

本研究で提案した緊急災害時制御モデルをハイブリッドクラウド環境に適用した際に, 仮想マシンインスタンスの物理的な配置を考慮した OpenFlow コントローラ制御の提案と実装を行い, その性能評価を行う. つまり, プライベートクラウド内に複製バックアップを行う際に, 仮想マシンインスタンスが物理的にどの Compute ノードに配置されているかを知ることができる管理者が緊急時に限り, OpenFlow コントローラを使って自動化した制御を行い, 重要なデータを扱うユーザのバックアップ処理を優先的に行う (図 7).

OpenFlow フレームワークの 1 つである Ryu を使い, 各 Compute ノード上にコントローラを起動させ, 各 Ryu コントローラがそれぞれの Compute ノード上で起動している仮想マシンインスタンスに繋がっている Open vSwitch をコントロールする. 各 Ryu コントローラは, それぞれ Compute ノード上で動いている仮想マシンインスタンスがどのユーザが使用している仮想マシンインスタンスかを学習し, 把握しているものとする. その上で管理者は, 緊急地震速報をトリガとしてプライベートクラウド内でバックアップ処理を迅速に行うために, それぞれの Ryu コントローラに自動的に指令 (クラウドコントローラ) を出し通信制御を行う.

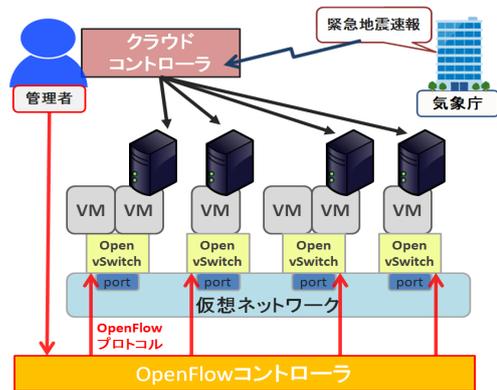


図 7 本研究の提案システムの概要

### 7.1 制御が想定される環境

今回, Ryu コントローラによる制御によるレプリケーション性能評価を行う際に想定される場合の一例を説明する. 図 8 で示すように, 各ユーザは仮想ネットワーク上の自分が権限をもつ仮想マシンインスタンス (以下, VM) のみ把握する. つまり, ユーザ A は VM1~VM6, ユーザ B は VM7~VM12 の権限をもつ. また, VM が物理的にどの Compute ノードに確保されるかは, スケジューラのアルゴリズム次第であり, 各ユーザは関知することができない. したがって, ユーザは知らないところ

で物理ネットワークでは影響を受けてしまう。図9のように、ユーザ A の通信がユーザ B の通信に影響を受けてしまうのである。

そこで、大地震といった緊急時にのみ管理者に全体のユーザの通信を制御可能とする権限を持たせ、各 Compute ノード上の上のっている Ryu コントローラに優先させたいユーザの通信を優先的に流すように制御させる。今回は、ユーザ A が重要度の高いデータを扱っているものとし、迅速にユーザ A のバックアップ処理を行うために他のユーザの通信を遮断させる。各 Compute ノード上の上のっている Ryu コントローラは、VM に繋がっている OpenvSwitch のポート VLAN の番号から、各 VM から流れるパケットをどのように処理するかを判断する。つまり本研究では、ユーザ B の VM に割り当てられたポート VLAN から流れてくるパケットにマッチするものは、パケットを捨てる (drop) 処理を行うフローエントリをフローテーブルに更新するアプリケーションが実行される。ここでいうフローエントリとは、OpenFlow でパケットをどう処理するかの情報のことであり、フローテーブルはこのエントリの保管庫のことである。

また、本システムで Ryu コントローラ (以下、Ryu) を各 Compute ノードに分散しているのは Ryu がスケーラビリティを考慮して作られているためである。Ryu は元々は集中型のコントローラであったが、大規模なシステムではそれがボトルネックになってしまう懸念があった。つまり、単純に実装するとコントローラの計算量が膨大になるため、スケーラビリティに問題が発生するからである。したがって、Ryu はバージョンアップに伴って分散型のコントローラに変更されており、各 Compute ノード上で動くことでコントローラがボトルネックになることを回避する。

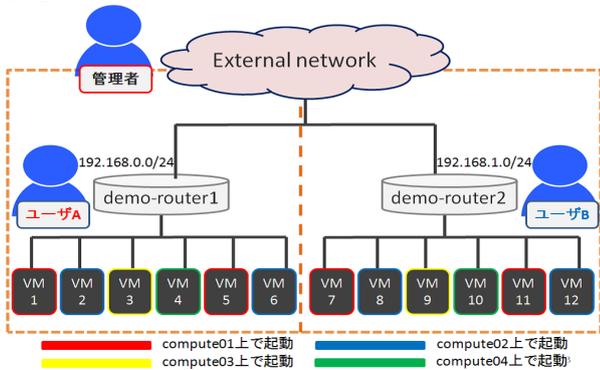


図8 仮想ネットワーク

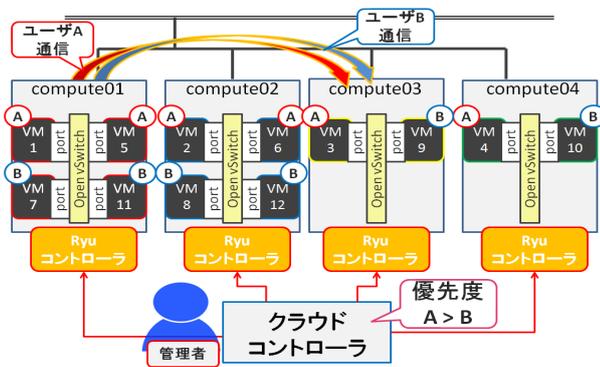


図9 物理ネットワーク

## 7.2 レプリケーション性能評価

各 Compute ノード上に Ryu コントローラを配置し、管理者による制御を行った場合にユーザ A がレプリケーションを行った際の実行時間とスループットを図10に示す。ユーザ B はユーザ A に影響を与える通信を発生させるために、ユーザ A がバックアップ処理を行う通信元・通信先の VM と同じように配置された VM 間でバックグラウンドトラフィックを発生させる。今回もネットワークトラフィックとして Iperf, ファイル転送として SCP を用いて負荷をかけた。異なるノード上に配置された VM にバックアップ処理を行う場合の方が、同じノード上に配置された VM にバックアップ処理を行う場合よりも実行時間が短く、スループットも比較的高い。Ryu コントローラで制御した場合、特に異なるノード上に配置された VM にバックアップ処理を行った際のスループットが大幅に上がり性能が向上したことが分かる。同一ノード上に配置された VM バックアップ処理を行った際には、Iperf の負荷をかけた場合とほぼグラフが重なっていることから影響を受けていないことが分かる。また、Ryu コントローラによる制御によってあまり性能が向上していないことからボトルネックはデータファイルへのアクセスのディスク I/O である。

この結果から、Ryu コントローラの制御時に大幅な性能向上がみられ、特に異なるノード上に配置された VM にバックアップ処理を行う場合に、迅速にバックアップを行えることが分かった

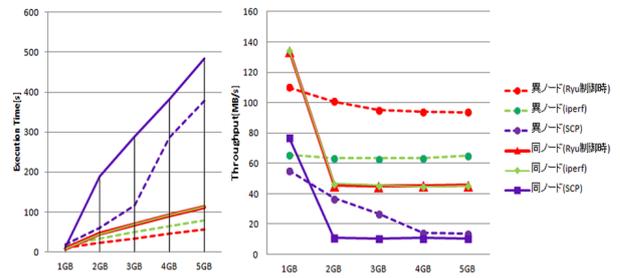


図10 実行時間/スループット

## 8. まとめと今後の課題

本研究は、近年大規模化・複雑化するハイブリッドクラウドにおいて、緊急災害時における迅速な対処に要する管理コストを抑えるための、自動的な制御とバックアップを行う提案システムを検討した。短い時間に複雑なシステムを制御する事は、人手では限界があるため、これを自動的に行う方式が重要である。そこで本研究では、実験環境であるハイブリッドクラウド環境を OpenStack を用いて構築し、ハイブリッドクラウド環境での緊急災害時制御モデルを提案し実装した。この緊急災害時モデルが、緊急地震速報などの大規模災害を引き起こす情報を入手した場合に、これに基づいてトラフィック制御を行う事により、重要なデータのバックアップ処理等が優先的に行われる仕組みについて検討を行った。また、震源地やマグニチュードの大きさからシステムに起きるであろう障害を予測し、それらの情報に応じて対処を行った。Ryu コントローラによる制御を

行いつつ、バックアップ処理をおこなうことは性能評価により有用であることが確認された。

今後は各 Compute ノード上でそれぞれ起動している Ryu コントローラがお互いに連携をとるような仕組みを検討したい。今回の実験における Ryu コントローラによる制御は、トラフィックを単純に止めた場合のみを検討しており、より複雑なシステムには対応しがたい。Ryu コントローラがお互いに連携を取ることができれば、より複雑なシステムにも様々な対応をすることができると考えられる。

## 文 献

- [1] 緊急地震速報について - 気象庁：  
<http://www.data.jma.go.jp/svd/eww/data/nc/>
- [2] 江丸裕教, 高井昌彰:「動的配置法によるハイブリッドクラウドの運用管理コスト最小化」情報処理学会論文誌, Vol.54, 1581-1591
- [3] OpenStack : <http://www.openstack.org/>
- [4] 石井秀治, 大山裕泰, 河合栄治 (2013) 「次世代ネットワーク制御技術 OpenFlow 入門」, アスキー・メディアワークス
- [5] "Open Networking Foundation":  
<https://www.opennetworking.org>
- [6] N. McKeown, et al. OpenFlow: Enabling innovation in campus networks, In Proceedings of SIGCOMM 2008, pp. 69-74, 2008
- [7] Ryu: <http://osrg.github.io/ryu/>
- [8] 飯島明夫「OpenFlow/SDN のキャリアネットワークへの適用について」電子情報通信学会ネットワークシステム研究会招待講演, 2012 年 10 月.
- [9] 多幡早紀, 堂ノ脇粹, 福井良太郎, 嶋津恵子, 重野寛:「OpenFlow を用いた災害時の動的な回線選択手法の検討」第 77 回全国情報処理学会, 2015 年 3 月.
- [10] 関野雄人, 柴田義孝, 内田法彦, 白鳥則郎:「OpenFlow をベースとした災害情報ネットワークにおけるリンク切り替え技法の実現に関する研究」, 研究報告コンピュータセキュリティ, 2013 年 3 月
- [11] 原瑠理子, 小口正人:「緊急地震速報に基づくハイブリッドクラウドにおけるバックアップシステムの検討」, DEIM2015, E1-2, 2015 年 7 月