

緊急地震速報に基づいたハイブリッドクラウドにおける OpenFlowによるネットワークトラフィックの最適化

原 瑠理子[†] 三島 健[†] 小口 正人[†]

[†]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, ^{††}mishima.takeshi@lab.ntt.co.jp, ^{†††}oguchi@computer.org

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、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

An Optimization of Network Traffic using OpenFlow in a Hybrid Cloud Based on Earthquake Early Warning

Ruriko HARA[†], Takeshi MISHIMA[†], and Masato OGUCHI[†]

[†]Ochanomizu University 2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610, JAPAN

E-mail: [†]ruriko@ogl.is.ocha.ac.jp, ^{††}mishima.takeshi@lab.ntt.co.jp, ^{†††}oguchi@computer.org

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、携帯型デバイスの発展、クラウドやデータセンタの増加により、いわゆる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[1]。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。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することが期待される。

また、クラウドやデータセンタにおいてはシステムの負荷変動に対し、仮想マシンの動作規模を変えたりマイグレーションを行うなどして、負荷変動の吸収を図る。現在、ネットワークをソフトウェアによって柔軟に制御可能な技術(SDN:Software-Defined Network)として注目されているOpenFlowを利用し、

ネットワークのトラフィック量の変動に応じて、クラウド間やクラウド・クライアント間を接続するネットワークの構成や帯域を可変(プログラマブル)なものとする検討が進んでいる[2]。しかしこれらの技術は、一般的に緩やかな負荷変動を念頭に検討されており、そのままでは短時間で激しく変化するバースト的な負荷変動に耐えることができない。

そこで本研究では、このようなバースト的な不可変動を、緊急地震速報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。緊急地震速報は、法人が財団法人気象業務支援センターと契約することにより受信できるようになるが、Twitter社のStreaming APIを使ってツイートとして流されており、これを受信するとほぼリアルタイムに情報を把握することができる。この情報を元に緊急時の状態を判断して、システム制御を行う。本研究ではこれを多くの企業で普及し始めているハイブリッドクラウド環境へ適用することにより、クラウドの安全性・通信性能の向上を図る。

2. Software-Defined Network

2.1 プログラマブルネットワーク

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[3][4]。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ような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。

2.2 OpenFlow

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。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。本研究では、外部情報から負荷変動を予測しOpenFlowを用いてコントロールすることで、不測の事態への対応を行う。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[5]。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。

2.2.1 OpenFlow コントローラ機能

OpenFlowでは図1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。

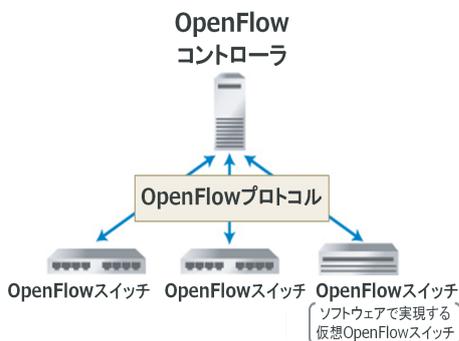


図1 OpenFlow コントローラによる制御

2.2.2 既存ネットワーク機器との連携

既存ネットワークにおいて、すべてのネットワーク機器を一斉に新たなOpenFlowスイッチに置き換えるのは困難であった。

しかし、OpenFlowスイッチは既存のネットワーク機器と同様に、さまざまな種類のデータを転送することができ、既存のネットワーク機器との混在が可能である。また、既存のネットワークの増設時や設定管理が複雑な機器などを対象に、部分的・段階的にSDNのアプローチでOpenFlowスイッチを導入していくことができる。Ryuでは、このような既存のネットワークの中に、OpenFlowスイッチを導入していくための機能を持っている。次節以降で述べるクラウド環境においても、OpenFlowによるネットワーク管理機構が浸透し始めている。

3. クラウド環境におけるシステム制御の課題

近年、クラウドコンピューティングの普及に伴い、クラウド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たハイブリッドクラウドが注目されている。パブリッククラウドは必要に応じてリソースを確保することができ、拡張性に優れている。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。

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。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。

例えば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。

4. 緊急災害時におけるシステム制御の提案モデル

4.1 制御モデルの提案

これまでの研究では、ネットワーク環境においてOpenFlowを使って制御する以下の3つのシナリオを提案し、Tremaの仮想システム環境[6]に実装して動作を確認した[7]。

- (1) ネットワークの切り替え.
- (2) 最短経路以外の最適な経路探索.
- (3) 複数経路を使って帯域の確保.

気象庁が Twitter 上で発信している緊急地震速報 [8] をモニタリングし [9], そのモニタリング情報をトリガとして, 3つのシナリオが実行されるコントローラを実装した.

(1) ネットワークの切り替え

災害によって, ネットワークの途中の通信経路が欠損・断続的切断が生じるような劣悪な環境に陥っても, モニタリング情報を元に臨時のネットワークに瞬時に切り替えることで, 緊急災害時の通信を確保する.

(2) 最短経路以外の最適な経路探索

緊急災害時には, 確実に届けたい情報 (安否確認など) がトラフィックに流れると想定されるので, 確実なネットワークを確保するために, 信頼性の高いパス (帯域が太い回線など) を選択してトラフィックを流す. もし選択された経路が, 最短パスでなかったとしても, 速さよりも確実性を重視する場合には, この手法は有用であると考えられる.

(3) 複数経路による帯域の確保

モニタリング情報によって, ある特定の地域で大きな被害が発生すると想定される場合, その地域から流れてくる大量のトラフィックを優先して流す必要があり, 通常のルーティングでは帯域の上限があると考えられるため, 同時に別の経路も使用してトラフィックを流すことで, 大幅に拡張した帯域を確保する.

4.2 緊急災害時フレームワーク

大地震や台風といった緊急災害などの, 短時間に起こる大きな負荷変動に情報処理基盤が耐えることは難しい. そこで, 大規模な災害をリアルタイムで検知し負荷変動を予測して, OpenFlow を用い, ビデオトラフィックなどの不要不急のトラフィックを制御して帯域の効率化を目指す「緊急災害時フレームワーク」を提案する. このフレームワークによって, 不測の事態への対応を行う.

本研究では投機的にトラフィックエンジニアリングを行う. もしかしたら結果的に, パースト的な負荷変動は起きないかもしれないが, 実際にシステムがダウンしてから復旧するには時間とコストがかかってしまう. 投機的にコントロールするコストよりも, 負荷変動が起きてシステムが耐え切れなくなった場合の損失の大きさは甚大であるため, パースト的な負荷変動が予測された時点で投機的にネットワークトラフィック制御を行う価値は, 社会インフラとしての情報基盤という観点から極めて高いと考えられる.

4.2.1 実装概要

Twitter 上で発信している緊急地震速報をトリガとして, 提案する「緊急災害時フレームワーク」に基づき投機的にネットワークトラフィックの制御を行う (図 2). 本研究ではさらにこれをクラウド環境にも適用し, 非常時にクラウドの制御が行われるようにする.



図 2 提案手法の流れ

5. OpenStack によるハイブリッドクラウド環境の構築

本研究では, 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 [10] を用い, 実験用のクラウド環境を構築した. 図 3 のように, コントローラノード 1 台, ネットワークノード 1 台, コンピュートノード 4 台の計 6 台からなるクラウド環境を 2 組構築した. そのクラウド間を人工的な遅延装置である Dummynet で繋ぐことで, 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構築している (図 4).

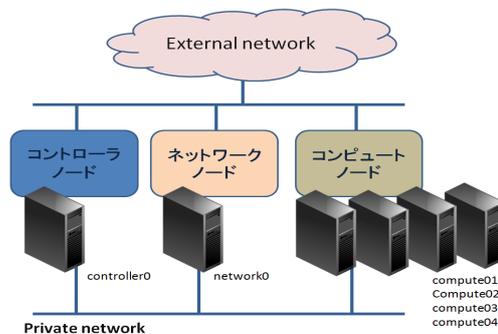


図 3 OpenStack によるクラウド構築

ハイブリッドクラウド

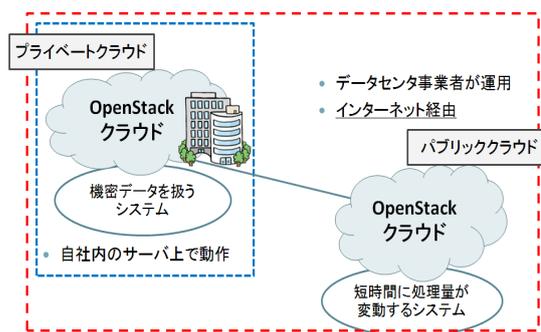


図 4 2つの OpenStack クラウドを接続したハイブリッドクラウド環境

OpenStack は, IaaS (Infrastructure as a Service) と呼ばれるクラウドサービスを提供する. OpenStack のユーザは, Web コンソールや API を利用して「仮想ネットワーク」「仮想マシンインスタンス」「ブロックボリューム」の 3 種類のリソースを中核とした, コンピューティングリソースをクラウド環境に用意

することができる (図 5)。これは物理的なマシンの境界にとられず構成された仮想的なクラウドシステム要素である。

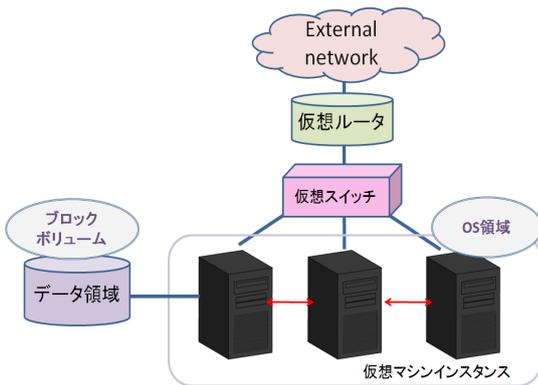


図 5 OpenStack で構築した仮想リソース

本研究の実験に用いた各ノードの性能を表 1 に示す。OpenStack のコントローラノード、ネットワークノード、コンピュートノードの全てに同一のマシンを用いた。クラウドのインスタンスはコンピュートノードの上で動作する。コンピュートノード間やインスタンス間の接続ネットワークを制御するのがネットワークノードであり、コントローラノードはクラウド全体を統括する。

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。本研究で用いた OpenStack のバージョン IceHouse は、図 6 に示すコンポーネントから構成される。各コンポーネントの機能については、表 2 の通りである。これらのコンポーネントは、1 台ないし複数台の物理的なマシン上で動作する。

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
Swift	分散オブジェクトストレージ
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
Cinder	ブロックストレージ
Ceilometer	リソース消費量を計測
Heat	オーケストレーション機能

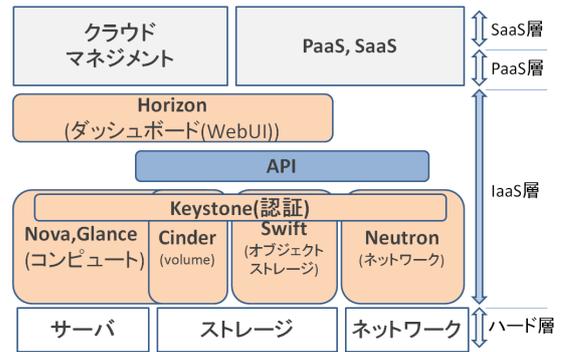


図 6 OpenStack コンポーネント構成 (IceHouse)

6. クラウド内でのレプリケーション性能評価

大規模な地震が発生した場合、急激なシステムへの負荷が起こったりサーバなどが物理的に被害を受けたりすることが考えられる。重要なデータが処理されているノードがダメージを負った場合、データの一部ないし全部が失われてしまう可能性がある。

そこで本研究では、重要度の高いデータを扱う可能性があるプライベートクラウド内において、地震発生直後にレプリケーションを行うケースを評価する。気象庁が Twitter で発信している緊急地震速報の震源地やマグニチュードの大きさの情報をトリガとし、重要なデータを確実にかつ迅速にレプリケーションするために、バックグラウンドで動いているトラフィックを制御して、制御前と制御後でのレプリケーションにかかる速度の性能評価実験を行う (図 7)。

また、Twitter 通知によって緊急地震速報を得てから実際にその地点で地震が発生するまでに 30 秒程度の時間があると仮定する。その場合、同じクラウド内で他のインスタンスにバックアップ退避をすることが可能なファイルサイズの大きさを計測する。

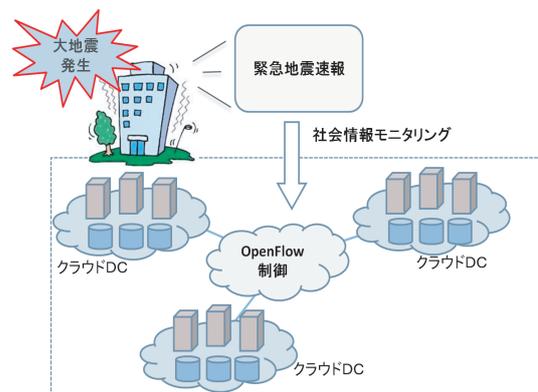


図 7 実験概要

6.1 実験概要

実験で用いた仮想システム環境は図 8 のように、仮想ルータ (demo-router) に仮想スイッチ (demo-net, demo-net2) を接続し、インスタンスを 4 つ起動させたものである。実験シナリオとし

では、このうち1つのインスタンスが重要なデータを処理しているものとする。緊急地震速報が送られてきたら、これをトリガとして、データファイルのバックアップ処理を開始する。バックアップのコピー先は異なるインスタンスとする。

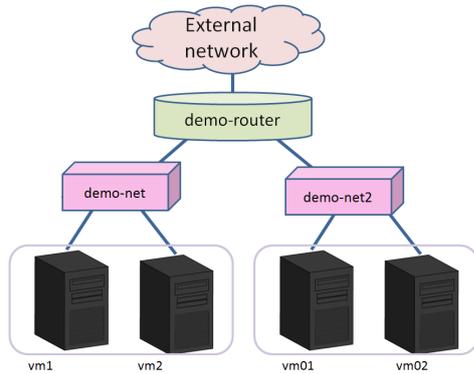


図8 仮想ネットワーク環境

これらのインスタンスとは別に、他のインスタンスはデータ処理や通信を行っており、これがネットワークやシステムのバックグラウンド負荷となる。緊急地震速報をトリガとして、重要なデータのバックアップを行う際に、バックグラウンドの負荷をそのままにした場合と、トラフィック制御によってバックグラウンドの通信を遮断し、バックアップのトラフィックを優先させた場合の比較を行う。バックグラウンドの負荷としては、インスタンス間で Iperf [11] を用いて通信トラフィックを発生させた場合と、バックグラウンドでもインスタンス間でファイル転送を行った場合の2つのケースについて比較した。

なお図8のインスタンスは仮想空間上に構築されるため、物理的にどの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、ユーザからは指定も関知もできない。OpenStackの管理者からは、どのインスタンスがどの物理ノードで動作しているかを確認することができる。本実験は、インスタンスが全て同一マシン上に配置されたケースと、インスタンスがそれぞれ異なるマシン上に配置されたケースの2つのケースで測定を行った。

6.2 性能評価実験

インスタンスが全て同一マシン上に配置された場合の実験結果を図9と図10に示す。図9はバックアップの実行時間、図10はスループットである。図9、図10とも通信制御時のグラフは、背景あり(Iperf)のグラフとほぼ重なっている。

これらの図からわかるように、バックグラウンドトラフィックとしてファイル転送(FTP)が行われているケースでは、そのままバックアップ処理を行う場合と比較して、トラフィック制御を行いバックアップ処理を優先させた場合には、大幅に性能向上する事がわかる。ただしバックグラウンドが Iperf による通信トラフィックのみである場合には、性能は制御を行ってもほぼ同程度であった。このことから、実際にボトルネックになっているのはネットワークではない事が予想される。

この場合、バックアップの性能を制約しているのは、データファイルへのアクセスのディスク I/O であると考えられる。ファ

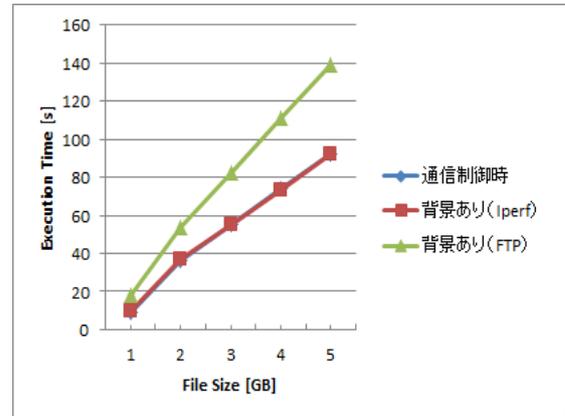


図9 バックアップの実行時間(同ノード)

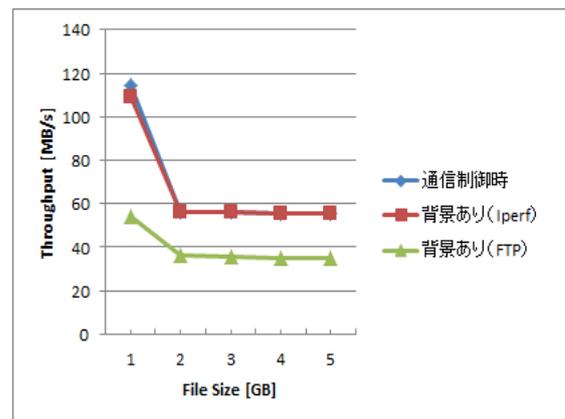


図10 バックアップ時のスループット(同ノード)

イルサイズが小さい時は、メモリ上にある程度載るので高いスループットが出るが、ファイルサイズが大きくなると飽和している様子からもその事がわかる。すなわちバックグラウンドの通信を遮断したため、そのジョブの I/O も止まり、結果としてバックアップ処理の性能が向上したと考えられる。

次に、インスタンスがそれぞれ異なるマシン上に配置された場合の実験結果を図11と図12に示す。

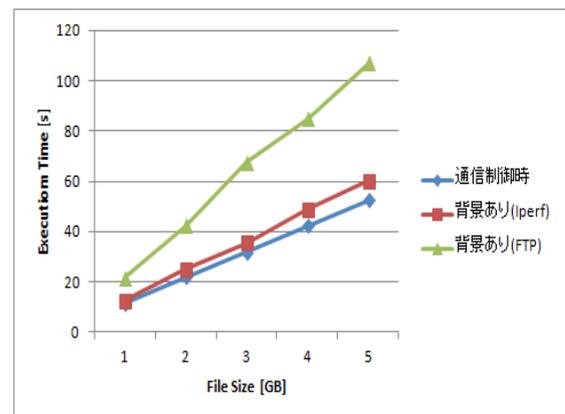


図11 バックアップの実行時間(異ノード)

図11と図12は同一マシン上に配置された場合の実行時間(図9)とスループット(図10)比較して、バックグラウンドで Iperf による通信トラフィックを発生させた場合にも、少し影響

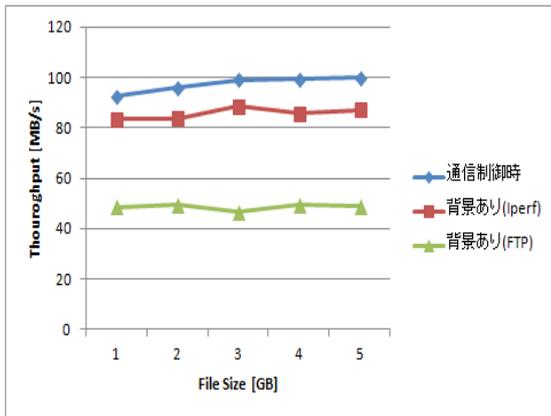


図 12 バックアップ時のスループット (異ノード)

が出ていることが分かる。これは、異なるマシン上に起動しているインスタンス間で実験を行っているため、同一マシン上での実験結果とは異なり、ネットワークの帯域に余裕がないことが要因であると考えられる。つまり、同一マシン上に配置された場合と比べて帯域が小さいため、バックグラウンドで FTP が流れるケースほどではないが、帯域が Iperf による影響を受けることになる。

また、クラウド内の他のインスタンスへのバックアップ退避時間の猶予が 30 秒以内である場合、図 13 で示すように 1.6GB 程度までのファイルサイズがバックアップ可能限度であることが分かる。この図のように、通信制御前に比べて大きなファイルサイズのバックアップが可能となった。

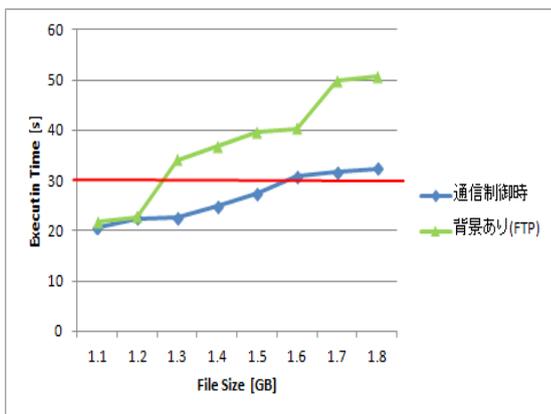


図 13 通信制御時のバックアップ実行時間

今回の実験で、同一マシン上に配置された場合、インスタンス間のバンド幅が大きくなるため、ディスク I/O 側がボトルネックになったが、インスタンスが異なるノード間に跨る場合には、ネットワークによる影響も受けることが分かった。

いずれにしても、図 9 や図 13 に示されるように、緊急地震速報がトリガとして入ってきてからの 30 秒以内に通信制御を行うことで、1.6GB 程度のファイルサイズまでバックアップを行うことが可能になると分かった。できるだけ多くのバックアップを行いたいと考えられる中で、少しでも短い時間に多くのデータのバックアップを取る事ができる方式は有効であるといえる。

7. まとめと今後の課題

本研究は、近年大規模化、複雑化するクラウドシステムにおいて、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。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。本研究では実験環境として OpenStack を用いてクラウドを構築し、OpenFlow のスイッチを用いたトラフィック制御の仕組みを導入した。バックグラウンドの処理を想定し、これを制御した場合としない場合の比較を行い、提案アプローチが有望である事を示した。

現在はインスタンス間の処理のみを検討しており、環境に応じて何をどう制御すべきか判断していく必要がある。今後はクラウド間へと検討を広げていきたい。特に提案手法は、例えば緊急災害時にダメージが予想される地域のプライベートクラウドから、遠隔のクラウドへデータのバックアップを行うような場合に最も有効であると考えられる。今後は実装を進めて、そのような場合の評価を行っていく。

文献

- [1] McKinsey Global Institute “Big data: The next frontier for innovation, competition, and productivity” June 2011.
- [2] 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.
- [3] “Open Networking Foundation”:
<https://www.opennetworking.org>
- [4] N. McKeown, et al. OpenFlow: Enabling innovation in campus networks, In Proceedings of SIGCOMM 2008, pp. 69-74, 2008
- [5] Ryu: <http://osrg.github.io/ryu/>
- [6] Trema: <http://trema.github.io/trema/>
- [7] 原瑠理子, 長谷川友香, 小口正人:「緊急地震速報に基づく OpenFlow を用いたトラフィックエンジニアリング」, 2014, 2G-5, 2014 年 3 月
- [8] 気象庁: <http://www.jma.go.jp/>
- [9] 長谷川友香, 小口正人:「緊急時判断に基づく状況に応じた個人情報へのアクセス制御」, DEIM2013, F5-6, 2013 年 3 月
- [10] OpenStack : <http://www.openstack.org/>
- [11] Iperf - The TCP/UDP Bandwidth Measurement Tool : <https://iperf.fr/>