

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

原 瑠理子¹ 小口 正人¹

概要：

近年、データセンタ事業者などが提供するパブリッククラウドと、企業が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、大地震のような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。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。つまり、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。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。

A Study on Backup System in a Hybrid Cloud based on Earthquake Early Warning

RURIKO HARA¹ MASATO OGUCHI¹

1. はじめに

災害時において、システムを安全に稼働させることは重要である。東日本大震災では、通信の途絶や停電によるシステムダウンが、情報共有に大きな支障をきたした。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。

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

ラムブル)なものとする検討が進んでいる[1]。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れば、迅速かつ安全に重要なデータをバックアップ処理できるのか検討する。

緊急地震速報[5]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

¹ お茶の水女子大学
2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 JAPAN

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 [2][3]。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 [4]。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。

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

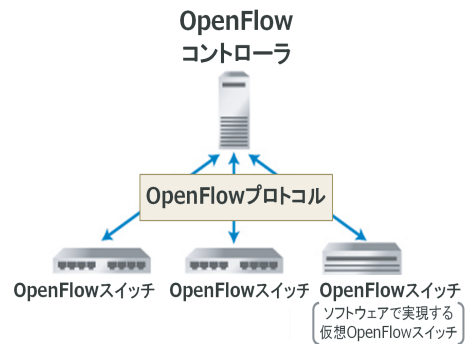


図 1 OpenFlow コントローラによる制御

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. 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 [6] を用い、実験用のクラウド環境を構築した。図 2 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模擬している。

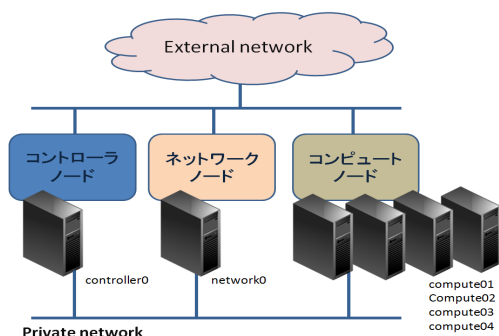


図 2 OpenStack によるクラウド構築

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。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

5. バックアップ性能評価

5.1 クラウド内でのレプリケーション性能評価

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した。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。

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。

5.1.1 実験概要

実験で用いた仮想システム環境は図 3・図 4 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。実験シナリオとしては、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。

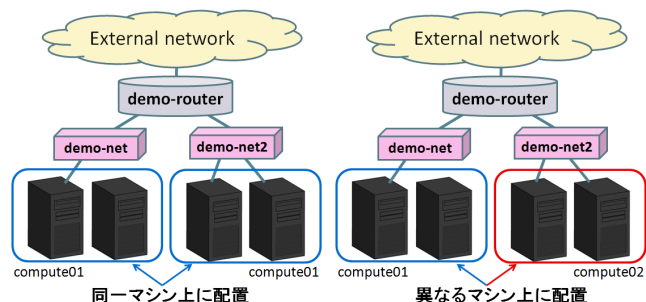


図 3 同ノードに配置された場合 図 4 異なるマシン上に配置された場合

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[7] 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(3)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(4)の2つのケースで測定を行った。

5.1.2 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5と図6に示す。図5はバックアップの実行時間、図6はスループットである。図5、図6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。

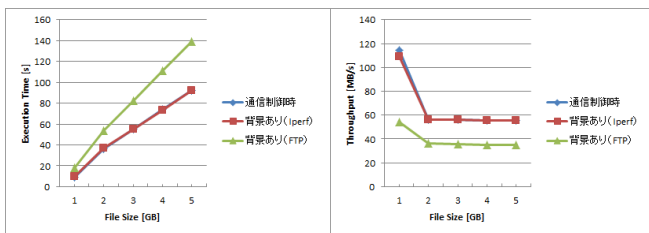


図5 実行時間(同ノード)

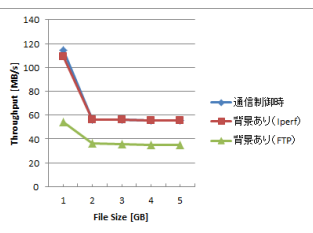


図6 スループット(同ノード)

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7と図8に示す。

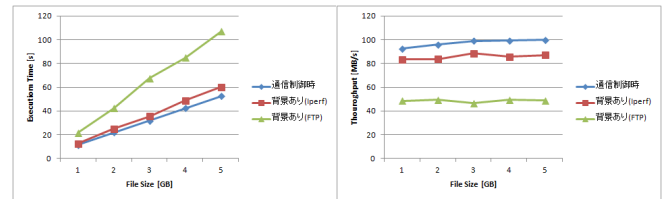


図7 実行時間(異ノード)

図8 スループット(異ノード)

図7と図8は同一マシン上に配置された場合の実行時間(図5)とスループット(図6)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。

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。

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、制御前は 1.25GB 程度のファイルサイズしかバックアップできなかったが、制御後は 1.6GB 程度までの大きさのバックアップが可能となった。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。

5.2 クラウド間でのレプリケーション性能評価

もし仮に、プライベートクラウドに物理的な障害が発生することが予測される場合、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ることで被害を最小限に抑えることができる。今回は、緊急地震速報の震源地の情報をトリガとし、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、重要なデータをもつクラウドから別のクラウドにバックアップ退避させた場合の制御を提案した。

5.2.1 実験概要

実験環境として、図9のように6台のサーバからなるクラウド環境2組を人工的な遅延装置である Dummynet で繋ぐことで、オンプレミスのプライベートクラウドと地理的

に遠い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。

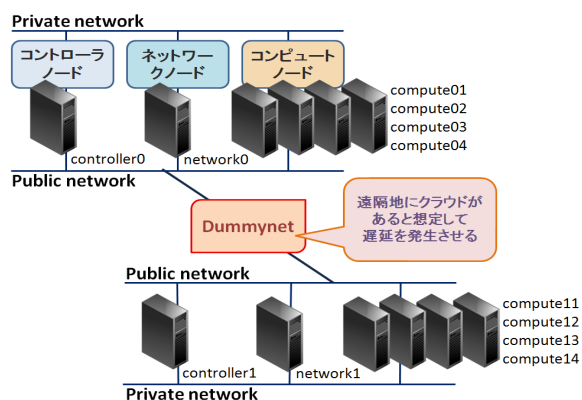


図9 クラウド間でのレプリケーション

6. まとめと今後の課題

本研究は、緊急地震速報の情報をもとに投機的にトラフィック制御を行うことで、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は重要であるといえる。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。またインスタンス間の処理に加え、クラウド間でのバックアップ処理についても検討を行った。

現在は、バックグラウンドのトラフィックを単に止めただけの制御しか行っていないため、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。今後は実装を進めて、そのような複雑な制御を検討し評価を行っていく。

参考文献

- [1] 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.
- [2] "Open Networking Foundation": <https://www.opennetworking.org>
- [3] N. McKeown, et al. OpenFlow: Enabling innovation in campus networks, In Proceedings of SIGCOMM 2008, pp. 69-74, 2008
- [4] Ryu: <http://osrg.github.io/ryu/>
- [5] 気象庁: <http://www.jma.go.jp/>
- [6] OpenStack: <http://www.openstack.org/>
- [7] Iperf - The TCP/UDP Bandwidth Measurement Tool : <https://iperf.fr/>