Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 98 (2016) 515 – 521
Procedia Computer Science

The 3rd International Symposium on Emerging Information, Communication and Networks (EICN 2016)

A Speculative Control Mechanism of Cloud Computing Systems based on Emergency Disaster Information using SDN

Masato OGUCHI[a,*], Ruriko HARA[a]

[a] *Ochanomizu University, 2-1-1 Otsuka Bunkyo-ku Tokyo, 112-8610, Japan*

Abstract

In recent years, a Cloud computing system has been popularly used. Among them, a hybrid Cloud is focused, which combined a public Cloud operated by service providers and a private Cloud constructed inside a company. Because a private Cloud is considered to be secure and a public Cloud is scalable, it is possible to build a Cloud computing system that works effectively by combining the both types of Clouds. However, when a big disaster occurrs, huge volume of data is produced by monitoring censors and users, and flowed into such an information infrastructure. In addition, enormous number of people access to the system in such a case, thus a load of the system becomes extremely high in a short time. Therefore, it is important to change an environment dynamically among inter-Cloud and intra-Cloud to deal with the load. In this paper, bursty increasing load is predicted based on Earthquake Early Warning (EEW), and a speculative control is performed in a short time between a time of occurrence of an earthquake and a time when the system is heavily loaded actually. In addition, network traffic is controlled to give a higher priority to important data replication. The system is constructed on a Cloud computing platform using public domain software, OpenStack in this experiment. Network traffic of the system is controlled by Software Defined Network (SDN), which is controlled by the OpenFlow protocol. As a result of an evaluation, our proposed system works fine and achieves good performance.

Peer-review under responsibility of the Program Chairs

Keywords: Cloud Computing, Disaster Information, Earthquake Early Warning, Speculative Control, OpenStack, Software Defined Network, OpenFlow

1. Introduction

Currently, a Cloud computing system is widely used. A hybrid Cloud is drawing an attention, which combines a public Cloud and a private Cloud. Because a private Cloud is considered to be secure and a public Cloud is scalable, it is possible to build a Cloud computing system that works effectively by combining the both types of Clouds.

However, when a big emergent event such as a large earthquake occurs, huge volume of data is produced and come into such an information infrastructure. When large-scale natural disasters occur, huge volume of data is coming into the system not only from monitoring censors but also from people through social media. In addition, because

* Corresponding author.
E-mail address: oguchi@is.ocha.ac.jp

Peer-review under responsibility of the Program Chairs
doi:10.1016/j.procs.2016.09.065

enormous number of people access to the system in such a case, a load of the system becomes extremely high in a short time. For example, when the Great East Japan Earthquake occurred in March 2011, the Web server of Fukushima Prefecture was down by too much accesses from people inside and outside of the prefecture to search for information. Almost ten times accesses of a normal level were recorded in that occasion.

It is difficult to reconfigure the system by hand in such a case. Therefore, it is important to change an environment dynamically among inter-Cloud and intra-Cloud to deal with the load. Important data should be backed up in a short time, for example, from a private Cloud to a public one. The information infrastructure is expected to continue to work and provide information even in case of large-scale natural disasters. In this paper, bursty increasing load is predicted based on Earthquake Early Warning (EEW) [1], and speculative control is performed in a short time between a time of occurrence of earthquake and a time when the system has a heavy load actually. Important data is replicated from a private Cloud to a public one in such a case. In addition, network traffic is controlled to give a higher priority to important data replication.

The system is constructed on a Cloud computing platform made of public domain software, OpenStack in this experiment. In an emergency case, a network should be directly controlled and reconfigured immediately. For that purpose, Software Defined Network (SDN) is used and the network is controlled by OpenFlow protocol, in which an OpenFlow Controller can directly send a command to OpenFlow switches to control the network [2]. As a result of an evaluation, our proposed system works fine and achieves good performance.

The remainder of the paper is organized as follows. A Cloud environment is explained and a speculative backup is proposed in Section 2. An experimental Cloud environment is introduced in Section 3. Controlling traffic of a Cloud network is described and the proposed mechanism is evaluated in Section 4. A conclusion and a future work are mentioned in Section 5.

2. Clound Environment and a Speculative Backup

2.1. Hybrid Cloud

Recently, a hybrid Cloud is focused, which combined a public Cloud provided by data center operators and a private Cloud constructed inside a company. For example, a public Cloud such as Amazon Web Services [3] and Google Cloud Platform [4] is used when much volume of computing resources are required temporarily because its elastic feature is suitable for such purposes. On the other hand, a private Cloud can be operated securely because it is managed inside a company. Therefore, it is possible to build a Cloud computing system that works effectively by combining the both types of Clouds. A hybrid Cloud can be control flexibly based on requirements of its user.

2.2. Issues for a Hybrid Cloud in the Case of a Large Disaster

Despite the various merites of a hybrid Cloud, it is not yet introduced so popularly to a company. One of the reasons is its complexity. It is difficult to control a large complicated system manually. Although a control mechanism of a hybrid Cloud has been investigated [5], this cannot be applied in case of emergency, for example, when a load of the system becomes extremely high in a short time. When a big disaster occurs, huge volume of data is produced and come into such an information infrastructure. In addition, enormous number of people access to the system in such a case, a load of the system becomes extremely high in a short time. Therefore, it is important to reconfigure the system on a hybrid Cloud dynamically to deal with the load.

In this case, it is useful to establish a system which is controlled autonomously and automatically. For example, in case of emergency, important data should be replicated from a private Cloud to a public Cloud and/or a control system can be migrated to a different Cloud. Especially because a hyrid Cloud is normally distributed in a wide area, it is effective to transfer significant data from a hazardous area to a safety zone.

2.3. Capturing Information from Earthquake Early Warning (EEW)

EEW is issued in case of earthquake by Japan Meteorological Agency, when an initial P-wave is monitored near a seismic center of the earthquake. S-wave, a main shock of the earth the earthquake comes several tens of seconds later to the most of areas distant from the seismic center. After several minutes, huge volume of data and information must flow into the system from monitoring censors as well as people through social media.

In our experiment, EEW information is received through Twitter bot service[6]. This is captured using Twitter API provided by Twitter Developers[7]. A dummy Twitter account which emulates the same behavior with a case in a real earthquake is created and used for a test purpose.

This experimental system also works in a real earthquake case. A real EEW can be a trigger of a system control described in the next subsection.

2.4. Speculative Backup in a Cloud Environment

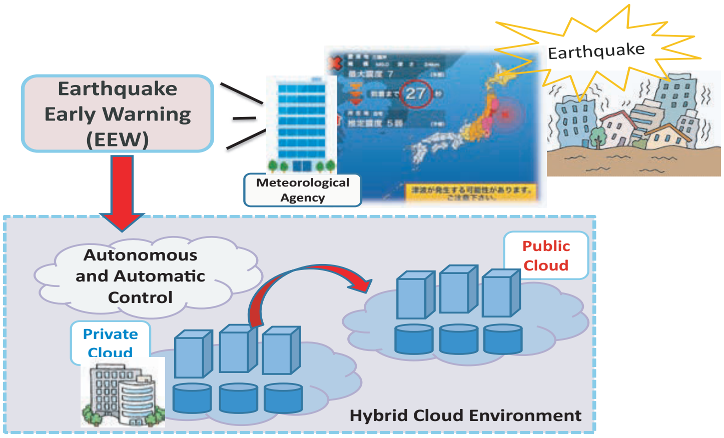

Fig. 1. An overview of the proposed system

When a large earthquake occurs in Japan, EEW is issued several tens of seconds before a main shock hits each area. Volume of data come in and out an information infrastructure is going to increase after several minites since then. Thus, it is possible to perform a speculative backup of significant data to prevent from its damage during such a short time among inter-Cloud and/or intra-Cloud. An overview of our proposed system is shown in Figure 1.

In this case, other network traffic should be suppressed because it decreases the backup performance. The network traffic of the speculative replication should be given the highest priority in the Cloud environment. This operation of the network is performed autonomously and automatically using SDN in our work, by developing controlling software based on OpenFlow protocol.

3. Experimental Cloud Environment

In this research work, a Cloud computing system of Infrastructure as a Service (IaaS) type is focused. OpenStack[8] is used for the implemetation environment in our experiment.

Our experimental system consists of two Cloud computing systems, each of them has one Controller Node, one Network Node, and four Compute Nodes. Figure 2 shows each Cloud computing system used for the experiment. The roles of each component of OpenStack is shown in Table 1.

The Controller Node manges the whole OpenStack system, and the Network Node controls network services connecting inside and outside of the system. The Compute Node provides a virtual machine (VM) called instance in OpenStack, using its physical CPUs, memories, and storages. Two Cloud computing systems are connected in this experiment and a hybrid Cloud are emulated consisting of a public Cloud and a private Cloud.

The specification of each node used for the Cloud computing system is shown in Table 2. The same type of nodes are used for the Controller Node, the Network Node, and the Compute Node.

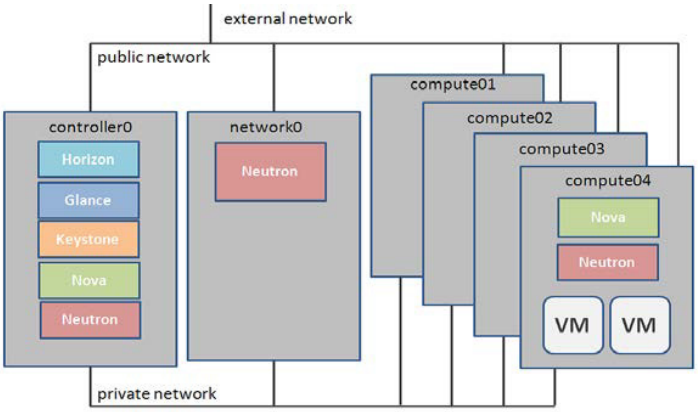

Fig. 2. Construction of an Cloud environment using OpenStack

Table 1. OpenStack Components （IceHouse）

Component	Role
Nova	Management of Virtual Machine
Glance	Management of Guest OS Image
Keystone	Integrated Authentication
Horizon	Web Console for Management
Neutron	Management of Virtual Network

Table 2. Specification of each node

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

4. Controlling Traffic of a Cloud Network

4.1. SDN/OpenFlow Framework

Because a network must be directly controlled and reconfigured immediately in the case of emergency, SDN is used and the network is controlled by OpenFlow protocol in this experiment. SDN is a programmable network system in which configuration, function, and performance of network can be controlled dynamically by software. OpenFlow is a representative protocol of SDN[9].

Among several implementations of OpenFlow frameworks, Ryu is used in our experimantal system[10], which includes OpenFlow controller as well as development tools and libraries for SDN applications. Although Ryu was originally a concentrated-type controller, it currently becomes a distributed-type controller which is intended to avoid to be a bottleneck of a system in a large-scale environment.

A control program of a network written by Python can be executed on a OpenFlow controller, which sends a control command to a OpenFlow switch. A software implementation of Open vSwitch[11] can be used as an OpenFlow switch. An overview of the proposed control system is shown in Figure 3.

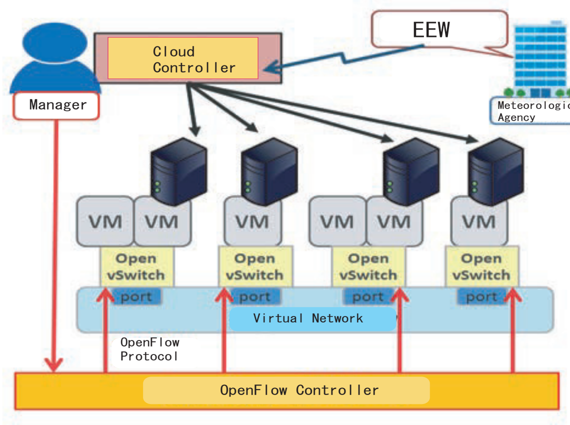

Fig. 3. An Overview of the Proposed Control System

4.2. *Traffic Control in the Case of Emergency on a Cloud Network*

Our proposed method to predict bursty increasing load and control network traffic is implemented on the experiment system described in the previous section. When the system receives an EEW signal, it triggers the speculative backup operation.

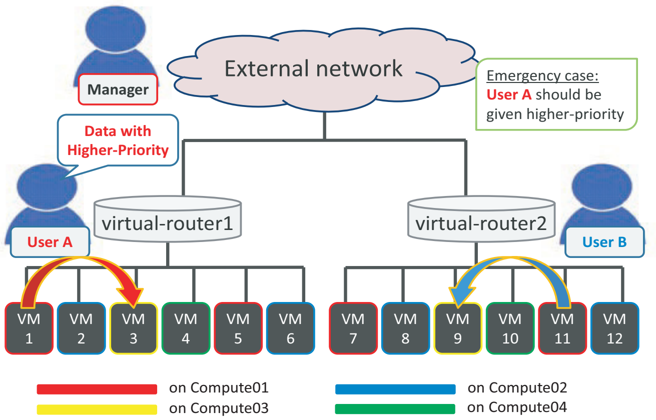

Fig. 4. Virtual configuration from the viewpoint of a Cloud user

A virtual configuration of a Cloud computing system is shown in Figure 4. In this case, for example, User A has data with higher priority, so that the data is replicated to another VM in case of emergency.

The network traffic of User B seems to be isolated from that of User A. However, this figure is illustated only from the viewpoint of a Cloud user. Cloud users can only view this image and cannot recognize where their VM is actually located.

On the other hand, a Cloud manager recognizes a physical configuration of the Cloud computing system as show in Figure 5. In this figure, it is shown that the network traffic of User A and that of User B use the same connection. Therefore, because the data of User A is higher priority, the traffic of User B should be suppressed in an emergency case. This mechanism is developed using SDN program in the Cloud computing system.

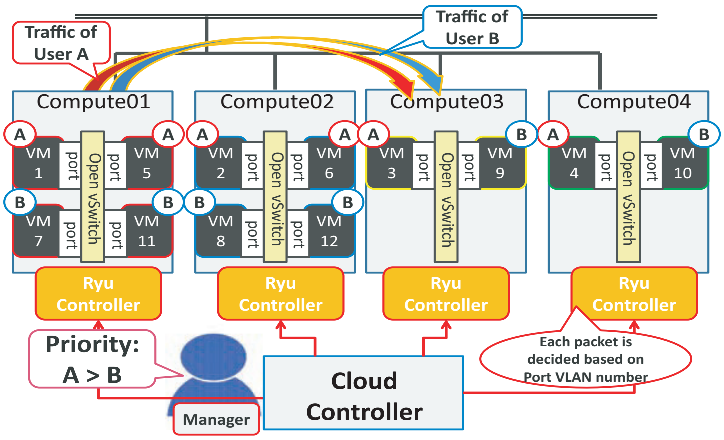

Fig. 5. Physical configuration from the viewpoint of a Cloud manager

4.3. Evaluation of Data Replication Performance using Traffic Control

The proposed and implemented mechanism of the speculative data replication and the traffic control is evaluated on the experiment system. Data replication performance with higher priority is evaluated when a EEW signal comes in. The data is replicated from one VM to another VM at the other Cloud. A background traffic exists on the same connection, like the traffic of User A and that of User B in Figure 5. The background traffic is controlled based on OpenFlow protocol using Ryu controller.

The following three cases are evaluated:

1. The background traffic is data copy (SCP), which is not suppressed
2. The background traffic is data transfer (iPerf), which is not suppressed
3. The background traffic is suppressed

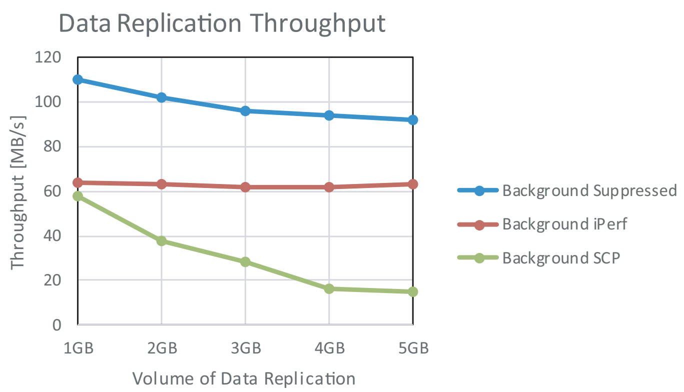

Fig. 6. Data Replication Throughput

The result of the evaluation is shown in Figure 6. The horizontal axis is the volume of replicated data with higher priority, and the vertical axis is throughput of replication.

Throughput of case 1 achieves relatively high throughput when the volume of replicated data is small. However, it decreases dramatically as the volume of replicated date becomes large, because a conflict on the same connection occurs.

In case 2, although a conflict should occur on the connection at the inter-Cloud network, throughput is higher than that of the case 1. According to these results, the bottleneck seems to exist at the I/O of storage for data copy, rather than the data transfer on the network.

When the background traffic is suppressed, in case 3, the highest throughput is achieved. As the result shows, our proposed mechanism of the traffic control in the case of the speculative data replication should be effective.

5. Conclusion

In this paper, a speculative control mechanism of a Cloud computing system is proposed and discussed. Important data is replicated among VMs in a hybrid Cloud computing system in emergency disaster case, triggered by EEW signal issued by Japan Meteorological Agency. A background traffic is suppressed that is conflicted with the traffic of important data replication, which is controlled using SDN program based on OpenFlow protocol. As a result of evaluation, the proposed mechanism to control a Cloud computing system should be effective.

As a future work, we will develop a more sophisticated Ryu program. Because Ryu is a distributed-type SDN controller, it is possible to develop a more effective mechanism that controls a Cloud computing system in emergency case.

Acknowledgements

We would thank Dr. Takeshi Mishima in NTT Laboratory for helpful suggestions and useful discussions for this research work.

References

1. Earthquake Early Warning (EEW), http://www.jma.go.jp/jma/en/Activities/eew.html (accessed April 25, 2016).
2. Open Networking Foundation, https://www.opennetworking.org/ (accessed April 25, 2016).
3. Amazon Web Services, https://aws.amazon.com/ (accessed April 25, 2016).
4. Google Cloud Platform, https://cloud.google.com/ (accessed April 25, 2016).
5. Evgeniy Pluzhnik, Evgeniy Nikulchev, and Simon Payain, "Optimal Control of Applications for Hybrid Cloud Services," IEEE 10th World Congress on Services (SERVICES2014), pp.458-461, June 2014, doi:10.1109/SERVICES.2014.88
6. Earthquake Early Warning Bot, https://twitter.com/eewbot (accessed April 25, 2016).
7. Twitter Developers, https://dev.twitter.com/ (accessed April 25, 2016).
8. OpenStack, http://www.openstack.org/ (accessed April 25, 2016).
9. Nick McKeown, et al., "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, Vol.38, No.2, pp.69-74, April 2008, doi:10.1145/1355734.1355746
10. Ryu SDN Framework, http://osrg.github.io/ryu/ (accessed April 25, 2016).
11. Open vSwitch, http://openvswitch.org/ (accessed April 25, 2016).