A Study about Web Application Inter-Cloud Auto-Scaling

Yuko KAMIYA, Toshihiko SHIMOKAWA

Abstract—Cloud Computing System is a basic infrastructure of current ICT systems. We propose "Meta-Cloud Computing". It makes it possible to use multiple Cloud Computing System as a single virtual Cloud Computing System. We develop a prototype system of Meta-Cloud Computing System. In this paper, we discuss about Meta-Cloud friendly server system configurations.

Index Terms— IaaS, Cloud Computing, Selection Policy

I. INTRODUCTION

Cloud Computing System, especially, IaaS, is a basic infrastructure of current ICT systems. Large number of systems are constructed on Cloud Computing System.

One of important feature of Cloud Computing System is Auto-scaling. It automatically launches or terminates virtual machines inside a Cloud Computing System based on pre-defined users' policies. It scales computational capability of servers depending upon actual workload.

There are large number of Cloud Computing Systems on the Internet. For example, EC2 of Amazon Web Services, Softlayer by IBM, Google Cloud Platform, and so on. These are so called "public cloud". On the other hand, users can build their own cloud so called "private cloud" by Open Source Software, ex. OpenStack, CloudStack, and so on.

There are some problems to construct system on Cloud Computing System. One of them is users have to select a base Cloud Computing System for their system, because there are very many Cloud Computing System as we mentioned above. The other one is network bandwidth for their system. Auto-Scaling can increase their computational capacity. However, it is difficult to increase network bandwidth. Auto-Scaling can increase virtual machine inside a Cloud Computing System. Even in that case, external bandwidth is not increased, because all of their virtual machines are inside same Cloud Computing System.

As we mentioned above, users have to select Cloud Computing System. To solve this, we propose "Meta-Cloud Computing" [1,2]. It makes it possible to use multiple Cloud Computing Systems as a single virtual Cloud Computing System. And we developed a prototype system of Meta-Cloud Computing System to assess the feasibility of Inter-Cloud Auto-Scaling. We call it "Soarin".

Many services are provided based on Web System. One of typical system architecture is Web 3-tier model. In this paper, we discuss about Meta-Cloud friendly system configurations.

II. META-CLOUD COMPUTING


"Inter-Cloud Auto-Scaling” has the other advantage. It can increase external bandwidth. As we mentioned above, it can increase virtual machine on the other Cloud Computing System. Therefore, it can use another external network between Cloud Computing System and the Internet. And also, it may decrease Round Trip Time between servers and clients.

A. Soarin: Prototype System of Meta-Cloud Computing

As we mentioned above, we developed a prototype system of Meta-Cloud Computing System “Soarin”.

To realize Inter-Cloud Auto-Scaling, Soarin has to be able to launch same virtual machines on multiple Cloud Computing Systems. If there are shared storage system, it is easy. Because, new virtual machine is able to launch from same HDD image as original virtual machine. However, this assumption is not correct on Inter-Cloud Auto-Scaling. To solve this problem, Soarin use configuration management tool to launch same virtual machine on different Cloud Computing System.

Many of servers are based on existing major Linux distribution, ex. Red Hat Linux, CentOS, Ubuntu, and so on. And, the servers use major server applications, ex. Apache, nginx, PHP, MySQL, and so on. Finally, their own application programs are installed on the server. On the other hand, almost

This work was supported by JSPS KAKENHI Grant Number JP26330124. Yuko KAMIYA and Toshihiko SHIMOKAWA are with the Kyushu Sangyo University, Fukuoka, JAPAN (e-mail: kamiya@is.kyusyu-u.ac.jp, toshi@is.kyusyu-u.ac.jp)
all Cloud Computing System has HDD image of unmodified major Linux distributions. Soarin launches virtual machine using these unmodified Linux distribution image. After that, Soarin uses configuration management tool to build up server on the virtual machine.


B. Cloud Selection Policies

Both of “Single Cloud Auto-Scaling” and “Inter-Cloud Auto-Scaling” launch or terminate virtual machines based on pre-defined users' policies. These policies contain metric and threshold to determine WHEN it launches or terminates virtual machines. In addition, policies of “Inter-Cloud Auto-Scaling” contains metric and threshold to determine WHICH CLOUD it launches or terminates virtual machines.

We call the policy “Cloud Selection Policy”. We implemented some basic Cloud Selection Policy as follows.

- Neighborhood of users
  RTT between Servers and Clients are important on interactive applications. This policy collects IP addresses of clients. It selects a Cloud Computing System depending on number of clients of each networks. Current implementation uses NetFlow [3] to collects clients’ information to reduce workload.

- Minimum resource usage
  Many of public Cloud Computing System provide very large computing resources. However, most of private cloud provide restricted computing resources. As we mentioned above, “Meta-Cloud Computing”

- Neighborhood of origin cloud
  Contents synchronization is one of important issue of distributed server. There are large number of solutions. This policy uses simple solution to reduce synchronization cost. It uses network distance between origin Cloud Computing System and each Cloud Computing Systems.

- Round Robin
  This is very simple selection policy. It selects a Cloud Computing System by round robin algorithm.

III. CONSIDERATION OF INTER-CLOUD AUTO-SCALING

In this section, we discuss about Inter-Cloud Auto-Scaling. First, we show some typical web based system architecture.

A. System configuration of some services

1) General web system architecture  (Web 3-tier model)

Fig. 1. shows general web system architecture (Web 3-tier model) including web server, application server, and database server. Auto-Scaling feature may launch new web server and/or application server to serve users’ requests. Local load balancer spreads these requests. In some cases, for example, the service many clients access database servers frequently or disasters happens, new database server is launched.

2) Video Streaming

Fig. 2. and Fig. 3. shows general video streaming architectures. There are two types of video streaming. Fig. 2. shows live streaming system. To deliver live streaming, encode server (Encoder) encodes a movie from video camera in real time. Streaming Server gets video data from encoder and delivers for users. Fig. 3. shows video on demand system. In this case, video data are prepared in advance. Therefore, real time encoding is not necessary.

Auto-Scaling feature may launch new streaming servers. In the case of live video streaming, contents synchronization is easy. On the other hand, in the case of video on demand, contents synchronization is big problem.

Fig. 1. General Web System Architecture (Web 3-tier model)

Fig. 2. Live Video Streaming System Architecture

Fig. 3. Video on Demand System Architecture
B. Inter-Cloud Auto-Scaling issue

1) General web system architecture (Web 3-tier model)

Inter-Cloud Auto-Scaling in the case of general web system architecture is useful. It can deploy new web servers in the Origin Cloud or in the different Cloud Computing System. If the different Cloud Computing System is chosen, the RTT between web servers and clients may become short compared with in the Origin Cloud. However, the RTT between servers and database servers should become long compared with in the Origin Cloud. To reduce the RTT, Inter-Cloud Auto-Scaling can deploy new database on the new Cloud Computing System. However, new problem: wide area database synchronization is arisen. RTT between clients and servers and database synchronization is tradeoff. It depends on services on the servers.

Session of web services also becomes problem. We will discuss it later at section C. Load Balancer.

2) Video Streaming

Inter-Cloud Auto-Scaling in the case of the live video streaming system is also useful. It can deploy new streaming servers in the Origin Cloud or in the different Cloud Computing System. In this case, contents synchronization is easy as we mentioned above. Therefore, it is easy to launch new streaming server on the other Cloud Computing System. We consider that Inter-Cloud Auto-Scaling is very useful for this service.

On the other hand, Inter-Cloud Auto-Scaling in the case of the Video on Demand have to consider about contents synchronization. Generally speaking, contents of video on demand is large. Therefore, it needs some contents synchronization mechanism.

C. Load Balancer

As we mentioned above, Meta Cloud Computing System is a natural extension of Cloud Computing System. Single Cloud Auto-Scaling increase servers inside a cloud. Inter-Cloud Auto-Scaling system increase servers across clouds. Auto-Scaling increases virtual computers automatically to scale out computing power. We faced new problem. We have to select computer to send request.

LLB: Local Load Balancer is an answer of this problem. Some of LLBs act as proxy server. It hides computers increased by auto scaling. Users do not need to know how many computers behind LLB. Users only need to send request to the LLB. LLB distributes request to suitable computers.

On the other hand, GLB: Global Load Balancer was used on widely distributed systems. ex. CDN: Contents Delivery Networks. CDN deploy their servers on the Internet widely. GLB also hides these servers as LLB does.

Load balancing mechanisms are difficult between LLB and GLB. Servers behind LLB are located nearly. Therefore, proxy server architecture is suitable. However, Servers behind GLB are located distributed. Therefore, most of GLBs used DNS mechanism and/or routing mechanism.

It is easy in principle to inspect request packet in LLB. However, it is no easy in GLB. DNS server cannot inspect request packet from client to server. It can inspect DNS request only. Routing mechanism cannot inspect any request packet. In addition, new servers are launched dynamically under Inter-Cloud Auto-Scaling. Therefore, result of Global Load Balancing may change dynamically. It is difficult to connect client to same servers to keep session. It is open issue.

IV. Conclusion

This paper shows concept of “Meta-Cloud Computing” and “Inter-Cloud Auto-Scaling”. It becomes able to use multiple Cloud Computing System as a single virtual Cloud Computing System. It is natural extension of Cloud Computing System and Auto-Scaling. We discuss issue about using Meta-Cloud Computing for existing internet based systems.

We have to implement solution of these issue into our prototype system Soarin, and evaluate it in the future.

REFERENCES


Yuko Kamiya received the B.S., M.S. and Ph.D. degree in computer science from Kyushu Sangyo University in 2006, 2008 and 2011. She is a assistant professor of information science in the Department of Information Science at Kyushu Sangyo University since 2014. Her research interests include wide area distributed computation, cloud computing, and e-education. She is a member of The Institute of Electronics, Information and Communication Engineers.
Toshihiko SHIMOKAWA received the B.E. and M.E. degree in computer science and communication engineering and the Ph.D. degree in computer science from Kyushu University in 1990, 1992 and 2001. He is a professor of information science in the Department of Information Science at Kyushu Sangyo University since 2007. His research interests include parallel and distributed computation, and wide area networking. He is a member of Information Processing Society of Japan and The Institute of Electronics, Information and Communication Engineers.