ISSN (Online): 1694-0814
www.IJCSI.org

Scalable Contents Delivery System with Dynamic Server Deployment

Yuko KAMIYA[1], Toshihiko SHIMOKAWA[2], Fuminori TANIZAKI[3] and Norihiko YOSHIDA[4]

[1] Graduate School of Information Science, Kyushu Sangyo University
2-3-1 Matsukadai Higashi-ku, Fukuoka 813-8503, Japan

[2] Faculty of Information Science, Kyushu Sangyo University
2-3-1 Matsukadai Higashi-ku, Fukuoka 813-8503, Japan

[3] NTT West Fukuoka Blanch
3-2-28 Hakataekihigashi Hakata-ku, Fukuoka 565-0456, Japan

[4]Information Technology Center, Saitama University
255 Shimooookubo Sakura-ku, Saitama 338-0825, Japan

Abstract

On providing broadband contents, to provide enough network bandwidth is an important. Existing Contents Delivery Network has mainly focused on increasing network bandwidth statically. Therefore, it is not flexible. In this paper, we propose *Soarin*, a novel contents delivery system to increase network bandwidth dynamically by deploying delivery servers in a wide area. Moreover Soarin can use various server deployment policy to deploy delivery servers, it can decide which server is suitable for content distribution. We call the criterion *server deployment policy*. We also propose several kinds of server deployment policies for typical contents delivery services.

Keywords: *CDN, Virtual Machine, Server Selection*

1. Introduction

With the rapid spread of the Internet, we can use broadband network even at home. Content holders provide several kinds of broadband contents for all over the Internet. They require network bandwidth to provide these contents. CDN (Contents Delivery Network)[1] is used to large-scale contents delivery. [1] describes that a CDN has some combination of a content-delivery infrastructure, a request-routing infrastructure, a distribution infrastructure, and an accounting infrastructure. The content-delivery infrastructure consists of a set of "surrogate" servers that deliver copies of content to sets of users. In this paper, we call the surrogate server *delivery server*. CDN can increase network bandwidth so that delivery servers are distributed in a wide area all over the Internet. Before using CDN to deliver contents, contents provider estimate the amount of the access to provision the enough processing power and network bandwidth. However CDN cannot provide their services during overload. This is because CDN cannot increase its network bandwidth and processing power flexibly.

Cloud computing makes it possible to increase processing power dynamically by increasing servers. However, current cloud-computing systems cannot increase network bandwidth. This is because it increases servers only in a local area. Servers have to be deployed in a wide area to increase network bandwidth.

There are three problems in deploying servers in a wide area. These problems is composed of three main parts:(1) how to deploy the servers, (2) where to deploy the servers, (3) when to deploy the servers. There are some research contributions for wide area live migration [2][3]. The results of these researches can be used in server deployment in a wide area. However, the purpose of these studies focuses how to deploy the servers. We tackle the problem when to deploy the servers.

In this research, we propose *Soarin*, a novel contents delivery system. Soarin can increase network bandwidth dynamically by deploying delivery servers in a wide area. Therefore Soarin is scalable. Moreover, Soarin can use various server deployment policies to deploy delivery servers. Therefore Soarin is *flexible*. As we mentioned above, there are still two problems about server deployment, *where* and *when*. Server deployment policy of Soarin is programmable, therefore Soarin can decide both of them using server deployment policy. Soarin selects a suitable physical machine and executes new delivery server inside a virtual machine on the physical machine. Generally speaking, a criterion of selecting physical machine is different by contents holder. We call the criterion *server deployment policy*. We also propose several kinds of server deployment policies for typical contents delivery services. Our contributions consist of the following:

- Propose an architecture of Soarin, a flexible and scalable contents delivery system.
- Propose some server deployment policy, which are suitable for typical contents delivery.

The remainder of the paper is organized as follows. Section 2 presents a related works, including an overview and problem of CDN. The proposed scalable contents delivery system: *Soarin* is introduced in Section 3 and Section 4 presents an evaluation of our system. Finally we conclude this paper in Section 5.

2. Related Works

Content Delivery Network (CDN) has been proposed for large-scale contents delivery.

2.1 CDN

CDN is widely used to large-scale contents delivery. CDN has some combination of a content-delivery infrastructure, a request-routing infrastructure, a distribution infrastructure, and an accounting infrastructure. The content-delivery infrastructure consists of a set of "surrogate" servers that deliver copies of content to sets of users. In this paper, we call the surrogate server “delivery server”. The delivery servers are deployed all over the Internet. These servers cache the contents from origin server using content-delivery infrastructure. The origin server stores the original contents. Requests from clients are redirected to their suitable deliver server by request-routing infrastructure. Finally, users retrieve contents from the delivery server by content-delivery infrastructure. Examples of commercial CDNs are Akamai [4] and Limelight [5].

2.2 Problems about Existing CDN

Existing CDNs cannot increase its network bandwidth dynamically. It is because most of them cannot increase delivery server dynamically. If content holders want to deploy new servers, they have a their own demand to select new location of servers. However, existing CDN provides only built-in rules that may not meet content providers’ demand. Therefore we can say that existing CDNs lack flexibility to deploy distribution servers.

3. Flexible Contents Delivery System with Dynamic Server Deployment

As mentioned 1.2., existing CDN lacks flexibility to increase network bandwidth. In this research, we propose “Flexible Contents Delivery System with Dynamic Server Deployment: *Soarin*”. We use *Server Proliferation* [6][7] as a basis of Soarin. Server Proliferation is introduced in 3.1.. 3.2 presents Soarin in detail.

3.1 Server Proliferation

Server Proliferation realizes increasing and decreasing processing power and network bandwidth of server system dynamically. To realize this, it increases and decreases servers in a wide area dynamically. Figure 1. shows architecture of Server Proliferation. We introduce two types of the servers in Server Proliferation. One is *Execution Server* (ES) and the other is *Distribution Server* (DS). Server Proliferation deploys physical machines that are installed virtual machine monitor all over the Internet in advance. These physical machines execute virtual machines on them. We call these physical machines *Execution Server*. The other server is *Deployment Server*. DS stores HDD images of virtual machines. In Server Proliferation, services (ex. Web server, Streaming server and so on) are executed inside virtual machines. When a new virtual machine is required, a HDD image of virtual machine is distributed from DS to one of the ES. The distributed virtual machine is executed on the ES. In this architecture, DS can become a bottleneck. However, it is easy to use multiple DS. Therefore, it does not become a bottleneck.

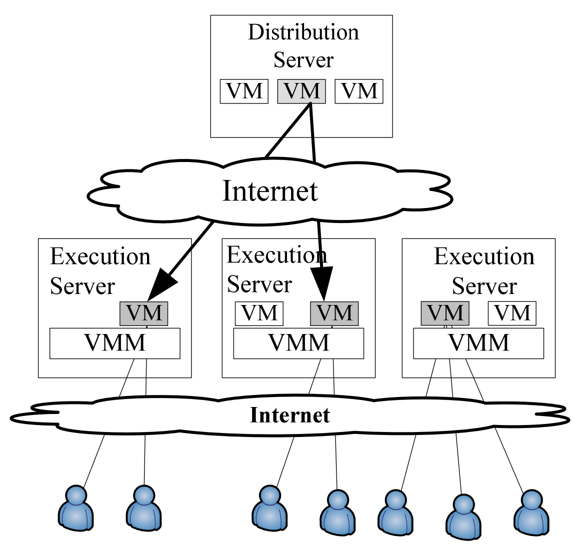

Fig. 1 Architecture of Server Proliferation

Cloud computing makes it possible to increase servers dynamically. By increasing servers, it is possible to use the CPU and the network of the increased server. Thus processing power and network bandwidth is increased. Therefore cloud-computing systems can increase the processing power and network bandwidth. However, compared with processing power, it is difficult to increase network bandwidth. This is due to network bottleneck. Typical cloud-computing system is constructed in an iDC (Internet Data Center). The uplink network of the iDC may become network bottleneck of the cloud-computing system.

By contrast, Server Proliferation can increase both of processing power and network bandwidth. It is because it can deploy servers in a wide area; therefore, deployed servers can use different uplink network each other. As it turned out, it is possible to increase network bandwidth of the system.

Server Proliferation uses virtual machine as a basis. It is because virtual machine is easy to increase and decrease dynamically. Moreover using virtual machine can reduce cost since physical machines can be shared with other system that uses virtual machines. It is possible to execute another virtual machine besides virtual machine executed by Server Proliferation. We can say that Server Proliferation is high-cost performance.

3.2 Soarin

In this section, we describe Flexible Contents Delivery System with Dynamic Server Deployment: Soarin. Soarin realize increasing network bandwidth flexibly. Soarin increases distribution servers to increase network bandwidth. Soarin can increase distribution server anytime. It is possible to add network bandwidth even after content distribution is started.

In addition, Soarin can decide when and where to increase distribution server flexibly. Upon increasing distribution servers dynamically, the new problem when and where to increase surrogates happens. In Soarin, distribution servers are constructed inside virtual machines, this problem is equal to a problem what criteria a physical machine to execute a virtual machine is chosen. We call this selection criterion *Server Deployment Policy*. We are studying about request navigation technology of CDN [8], and we clarify that there is a variety of selection criterion of request navigation. Hence it is obvious that there is a variety of server deployment policy, too. That is to say, server deployment policy is different by contents holder. Soarin can use various server deployment policies for contents holders' requirements. The examples of the server deployment policy are discussed in detail in section 3.3.

Soarin uses Server Proliferation to deploy distribution servers. Server Proliferation provides how to deploy a virtual machine dynamically. However it lacks capability to decide timing and location of deploying virtual machine. Therefore, we introduce *Observation Server* (OS) and *Control Server* (CS) in addition to *Execution Server* (ES) and *Deployment Server* (DS) in Server Proliferation. OS collects several kinds of metrics using server deployment policy. For example, it collects the CPU load and network traffic of ESs, calculates the distance from ESs to DSs. CS controls all over Soarin's system. CS selects ES to deploy new virtual machine using information from OS based on server deployment policy. Afterward CS directs DS to transfer HDD image of the virtual machine to the selected ES to deploy new distribution server. After that, CS directs ES to execute the virtual machine for new distribution server. Finally CS updates request navigation policy to use the new distribution server. Soarin uses *Tenbin*[8] as a request navigation system.

3.3 **Typical Server Deployment Policies of Soarin**

As mentioned above Soarin is able to use various server deployment policies. Server deployment policy may differ by each content holder. In this section, we show five typical server deployment policies. It is also possible to use other policies and/or combine these five server deployment policies.

1.Distance between Execution Server and Deployment Server

This policy uses the distance between Deployment Server and Execution Server as a criterion. Some metrics can be used to calculate the distance, number of hops, round trip time, and AS (Autonomous System) path length between DS and ES. DS measure these information periodically. OS collects these information from DS. This policy chooses the nearest ES from DS; therefore it may deploy new distribution server in a short time. As a result, it is possible to correspond to a sudden surge in the volume of request.

2.Processing Power of Execution Server

This policy uses processing capacity of Execution Servers as a criterion. OS collects processing power information of each ESs in advance. OS collects load averages of ESs periodically. OS calculates processing capacity of ESs with the load average and ESs' own processing power. Then OS chooses most powerful server. This rule can solve the lack of the processing power of distribution servers.

3.Network Bandwidth

This policy uses capacity of network bandwidth of ES as a criterion. OS collects network bandwidth of uplink of ESs in advance. OS collects usage of network bandwidth of uplink of ESs. OS calculates capacity of network bandwidth with these information. OS chooses the ES that has the broadest network bandwidth. Thus we solve a lack of network bandwidth.

4. Region where largest number of clients
This policy uses number of clients per region as a criterion. Soarin can deploy new distribution servers, however if the server is far from clients, the increased network bandwidth cannot use effectively. Therefore this policy tries to select nearest ES for most of clients. This policy has to define boundary of region. Some definition is available. Examples are Country, continent, and Autonomous System (AS)[9]. AS is a unit of inter-networking routing on the Internet, typically an Internet Service Provider or a very large organization. We can use IP address of clients to distinguish region. To distinguish the country from IP address, we can use information from RIR (Regional Internet Registry). A RIR is an organization that manages the allocation and registration of IP addresses with in a particular region of the world. There are five RIRs: ARIN, RIPE NCC, APNIC, LACNIC, and AfriNIC. ARIN manages North America, RIPE NCC manages Europe, the Middle East, and Asia, APNIC manages Asia and the Pacific area, LACNIC manages Latin America and the Caribbean Area, and AfriNIC manages Africa. We can recognize country of IP address from the RIR's information, and also recoginize that RIR assigns the IP address. If IP address of clients and ES are assigned from same RIR, it may exists near area. Generally speaking, the network bandwidth of the near area is wider. For example, it is rational to choose the ES on Europe but not on North America and or Africa toward the access from Europe.

5.Contents holder's opinion
This policy uses contents holder's opinion as a criterion. For example, if economical cost is not same between ESs, there is a possibility that not only the performance but also the cost is a better criterion. At that time the contents holder wants to choose ES whose costs are cheap. On the other hand, some content holder wants to do traffic engineering. Accordingly, the content holder selects ES by manual.

4. Conclusions

There are two problems in deploying servers in a wide area. One is where to deploy the servers, and the other is when to deploy the servers. In this paper we tackled the problem where to deploy the servers. In this paper, we proposed Soarin, a novel contents delivery system. Soarin can increase network bandwidth dynamically by deploying delivery servers in a wide area. Moreover Soarin can use various server deployment policy to deploy delivery servers, it can decide which server is suitable for content deliverer. We call the criterion "server deployment policy". We also propose several kinds of server deployment policies for typical contents delivery services.

Acknowledgments

This research was supported in part by MEXT in Japan under Grants-in-Aid for Scientific Research on Priority Area 21013008, and by JSPS in Japan under Grants-in-Aid for Scientific Research (B) 20300024.

References

[1] Day, M., Cain, B., Tomlinson, G. and Rzewski, P.: "A Model for Content Internetworking (CDI)", RFC3466 (2003)
[2] Takahiro Hirofuchi, Hirotaka Ogawa, Hidetomo Nakada, Satoshi Itoh, Satoshi Sekiguchi, "A Storage Access Mechanism for Wide-Area Live Migration of Virtual Machines", IPSJ SIG Technical Reports. [High Performance Computing] 2008(74), 19-24, (July, 2008)
[3] Robert Bradford, Evangelos Kotsovinos, Anja Feldmann, and Harald Schiberg, "Live wide-area migration of virtual machines including local persistent state. ", Proceedings of the 3rd international conference on Virtual execution environments, pp. 169–179(2007)
[4] Akamai Technologies Inc: Akamai; The Business Internet. http://www.akamai.com/
[5] Limelight: http://www.limelightnetworks.com/
[6] Yuko Kamiya, Toshihiko Shimokawa, "Scalable Server Construction Method Based on Virtual Machine Transfer and Duplication," Proceedings of International Multi-Conference on Engineering and Technological Innovation IMETI 2008, Vol.2, pp.273-278 (June, 2008)
[7] Yuko Kamiya, Toshihiko Shimokawa, Norihiko Yoshida, "Scalable Server System Based on Virtual Machine Duplication in Wide Area", Proceedings of The 3rd International Conference on Ubiquitous Information Management and Communication, pp.432-436 (January, 2009)
[8] Toshihiko Shimokawa, Norihiko Yoshida, Kazuo Ushijima, "Server Selection Mechanism with Pluggable Selection Policies", Electronics and Communications in Japan, Part □(Fundamental Electronic Science), Vol.89, No.8, pp.53-61(July, 2006)
[9] Rechter, Y., Li, T., "A border gateway protocol 4(BGP-4)", RFC1771(1995)

Yuko Kamiya received the B. Information Science and the M. Information Science from Kyushu Sangyo University in 2006 and 2008 respectively, where she is currently working toward the Ph.D degree. Her research interests include load balancing and wide area networking. She is a member of Information Processing Society of Japan.

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.

Fuminori Tanizaki joined NTT West in 1986. Since he joined NTT West, he has been engaged in Network & Server Operation. His research interests include IPv6, Video Streaming, DNS, and BSD UNIX.

Norihiko Yoshida received the M. Eng. degree in mathematical engineering and information physics in 1981 from the University of Tokyo, and the Dr. Eng. degree in computer science and communication engineering in 1990 from Kyushu University. He is Professor of computer systems at the Department of Information and Computer Sciences since 2002, and Director of Information Technology Center since 2008, both of Saitama University. His research interests include system design methodologies, and parallel and distributed computation. He has been served on program committees and organizing committees of various international conferences and symposia. He is a member of IPSJ, IEICE, JSSST, ACM and IEEE.