Lecture Notes in Electrical Engineering 376

Kuinam J. Kim
Nikolai Joukov
Editors

Information Science and Applications (ICISA) 2016

Springer

Contents

Optimized Data Transmission Method for Multimedia Services Based on OpenStack

Sanghyun Park, Gemoh Maliva Tihfon, Seonhyeok Lee and Jinsul Kim

Abstract In this paper, we provide optimum multimedia services, based on OpenStack VM. To provide optimal multimedia services when a user requests content, the network is configure to look for the fast server speed priority depending on the order in which the data can be transferred. In addition, based on the speed of the server providing the data sent is proportionally dividing, sending to the user its divide the physical resource based on a VM (Virtual Machine), it could use the resources efficiently. May be by classifying the managed contents using the VM, also has the advantage that the user can search quickly when they request for content. In this paper, an experiment was carried out, we apply the developed algorithm and we compare the network environment with the existing environment. We were able to demonstrate that the transfer of multimedia is fast and reliable than the conventional network environment.

1 Introduction

The growth rate of the network is rapidly evolving but to meet the satisfaction of those who demand it has failed. Providers have increase the number of servers to provide many services, however it is difficult to predict the size of a large amount of real-time network traffic and its dynamically changing traffic patterns. A number of studies have been conducted to address this issue. NFV is configured by hardware to the existing Route, NAT, Firewall, IDS, IPS, DNS, Caching, including controls by implementing various types of virtualization software capabilities to the device. If the number of the current service use NFV, they should be able to provide a personalized service according to various

S. Park · G.M. Tihfon · S. Lee · J. Kim(✉)
School of Electronics and Computer Engineering, Chonnam National University,
77 Yongbong-ro, Buk-gu, Gwangju 500-757, Republic of Korea
e-mail: sanghyun079@gmail.com, gemohmal@gmail.com, wingmanf1@gmail.com, jsworld@jnu.ac.kr

K.J. Kim and N. Joukov (eds.), *Information Science and Applications (ICISA) 2016*,
Lecture Notes in Electrical Engineering 376,
DOI: 10.1007/978-981-10-0557-2_124

environments by applying this technique. In this paper, we use the OpenStack based on an open platform, and offer the best data transfer method to provide multimedia service. We applied the data division method algorithm according to the network conditions to provide optimal multimedia services. In addition, the VM environment was configured to use the resources of the servers more efficiently. We analyzed the existing multimedia data server and the one we are proposing.

2 Related Work

Typically, they use a lot of the CDN server environment for data transmission method for dividing an optimal data transmission service. CDN origin servers in the environments are far away from the other server transfer efficiency. In this regard, a large number of research papers [1, 2, 3], are methods to set a routing path that has been considered by placing a replica for efficient multimedia transmission in the CDN environment. In consideration of the availability of the first replica is the fixed number. Cooperation with the proxy server using scheduling method for efficient multi-media transmission on the CDN environment is in paper [4]. Which is located near the client generates a request, select an optimized algorithm Prefix. [4] are multiple asynchronous clients proxy for optimizing the video stream from the server through the object and are, with the best Proxy sought to reduce the required network prefix cache allocation band. In addition, this approach was to reduce the transmission costs in a similar way presented in [5], a method was proposed to design a similar manner to the content distribution system which can be used to enhance the traditional approach. Therefore, in paper [5], we try to solve the clients' problems in order to collect the best real network topology. The server uses the VM for best utilization of the server data. Virtual machine system in previous research had to efficient use the resource scheduling algorithm. Each VM, the VM monitoring is provided with a different type of resource scheduling [6, 7]. [8] has considered the power efficiency of the green computing server to allocate resources dynamically in accordance with the request from the client in a cloud computing environment. This method is to measure the processing efficiency of the server in the multi-faceted resource using the method of 'skewness'. In order to transmit data in this way, it is best to study a variety of ways. We would like to provide the user with the multimedia data through the best-server environment. Furthermore, by checking the environment of the users in the network and the utilization of the server, and it provide the optimal multimedia services to users.

3 Division Transmission for Optimal Multimedia Services

In order to transmit from the optimal multimedia server, it is configured with Figure 1 in the form: Each of these servers are located in different locations, when a user requests a content, the Management Server serves to locate the requested content and provide the user with a multimedia service. The data server applies the choice algorithm shown in Table 1. If u user requests content data, as shown in Figure 1-①, the Management Server searches the entire server $s_{i,}$as shown in Figure 1-②. Next, the server stores the server $L_s(i)$ that matches the user's location u_{loc}in the entire server s_i. Included in Figure 1-③ the servers L_s check the responsiveness of the data around the user. It is arranged in ascending order from the high speed to low speed by the response time of the server. Management Server is the data server using the Sort (i) in ascending order to scan a server that has the content requested by the user. If the server has the content, the server in the Sort (i) is stored back to the $S_f(i)$. Figure 1-① ~ ③ provides users with multimedia services by applying the algorithm as shown in Table 1, in order to select the appropriate server in your environment.

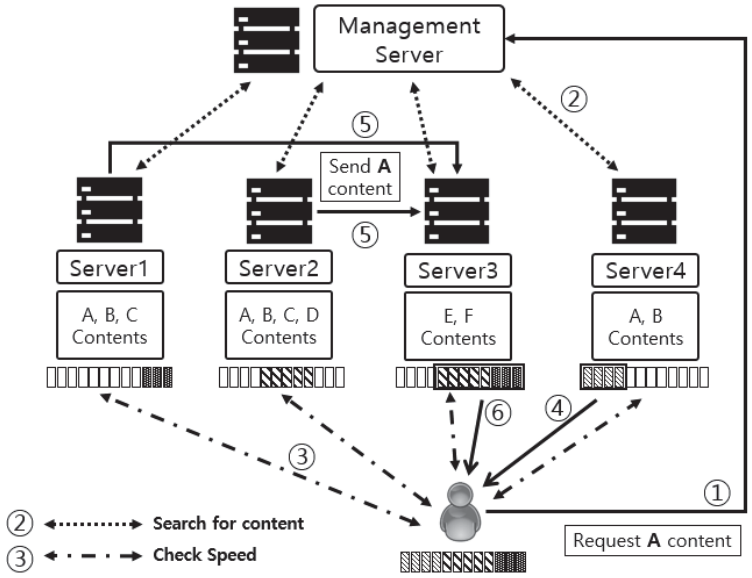

Fig. 1 Configuring the network environment

Algorithm 1. Searching Server for Multimedia services algorithm

WHEN u Request the Content Data **THEN**
//Management Server Connect using Http Protocol
WHILE S_i!= NULL **THEN** //Server Search
IF S_i== u_{loc} **THEN** // Server Location Check
$L_s(i)$ = S_i
END IF
END WHILE
WHILE $L_s(i)$!= NULL **THEN** //Response Speed Check
Sort(i) = $L_s(i)$; //Ascending Sort
END WHILE
WHILE Sort(i) != NULL **THEN**
IF u_{con} == Sort(i) **THEN**
$S_f(i)$ = Sort(i);
// Find the server containing the requested content
END IF
END WHILE
END WHILE

Table 2 shows the transmission of multimedia data segmentation algorithm according to the situation. The algorithm in Table 2 is applied to Figure 1-④~⑥. If the user requested content from the server $S_c(i)$ transmission is ready, the server checks the first fast server with content $S_c(0)$ of the fastest servers in the local server as the user $\mathrm{Sort}(0)$ with the same server. If it is such a server, the server will provide the requested content to the user. The server is different from each other such as Figure 1-④ server is the server with the fastest content, provides a streaming format data to the user first. The remaining servers can share the data to be transmitted to the user in a ratio-metric proportion with the speed and size. $S_f\ is$ the file size to be divided according to the ratio. 1 refers to the entire file. $S_f(0)$ need to provide the service to the user first, because it is directly connected to the user by the front. $S_b(i+n)$ divides the data of the user between the servers based on the speed ratio S_b belonging to $S_c(i)$, as in Equation 1, except for the server that is to transmitting directly. The remaining file size is the total data 1-$S_f(0)$. Next, each of the server transmission as shown in Figure 1-⑤, multimedia data is assigned to the fastest Sort(0) server. Table 2 is a multimedia data transmission or the $S_c(0)$ server, the best Quick Sort(0) on the server, when the data transfer is complete as shown in Figure 1-⑥, the user is connected to the fastest server, the user then receives the requested multimedia content.

Algorithm 2. Dividing the multimedia data transmission algorithm according to the situation

WHEN $S_c(i)$ Server ready for data transmission **THEN**

IF $S_c(0)\ !=\ \mathrm{Sort}(0)$ **THEN**

// If you do not have the fastest server Content

$S_c(0).send();$

// Earlier data transmission of the multimedia data

WHILE $S_c(i+1)$!= NULL **THEN**

// Data partitioning according to server ratio

$$S_f(i+1) = \left(1 - S_f(0)\right)\frac{s_b(i+n)}{s_b(i+1)+\ \ldots\ +s_b(i+n)} \quad (1)$$

$S_c(i+1).send();$

// Data transmission according to the ratio

END WHILE

END IF

IF $S_c(0) == \mathrm{Sort}(0)$ **THEN**

// If you have the fastest server content

$S_c(0).send();$

END IF

IF *(Complete* $S_c(0)$ *data transfer) or (Complete* Sort(0) *data transfer)*

THEN

Sort(0).Send(); // If you have the fastest server content

END IF

END WHEN

4 How to Use the VM Resource Utilization of the Server

Figure 1 is a OpenStack-based [9] server VMs, VM is generated in the Compute. Figure 2 shows each physical resources such as VMs and it is Management. Figure 2-① the dividing content on VM1 and VM2 depends on the size and number of content, to be shared. Figure 2-① Management is divided on the VM1 and VM2 in accordance with the contents of the capacity and frequency of use. For example if a user requests 'F' content, via the Management included in each physical resource makes a connection with the server that has the content corresponding to the user. If another user requests ‘B’ content, the Management

after searching the VMs with the 'B' content allows using the vSwitch connects the VM with the user and the content. Figure 2-① of the VM1 and VM2 in the resource utilization is only 30%. However, Figure 2-② utilization increase more and more as additional users want to use the 'F' content. Users gradually increase as the VM1 that has the content 'F' is to provide a service to the users, resource utilization increases more and more. Because of the VM resource utilization going beyond the maximum value of the function can cause it to stop, the VM Management should check the real-time utilization of the VMs. If it is determined that a particular VM resources are heavily used among the VMs, Management selects the VM that is using a lot of resources. And the content that the users have requested should be copy to a VM whose resource utilization is less. Then using the vSwitch, it allows the user to connect to the VM, copy the contents of the VM that is currently connected. Figure 2-③ illustrate there are groups of users that can see that changes of three user groups, two groups in Figure 2-②. Figure 2-② are users requesting the content 'F', etc. the utilization of the VM resource utilization rise to 60%. However, after copying the 'F', the content of VM1 to VM2, the Management uses the vSwitch to send users to go to VM1 and VM2. As shown in Figure 2-③ utilization of VM1 and VM2 is confirmed. Thus, by providing the configuration virtual resources and the general physical resources to the service environment using the VM, it is possible to speed up the existing multimedia control process. In addition, with a single server, by configuring the server in a VM environment, its resource can be effectively used as much as possible.

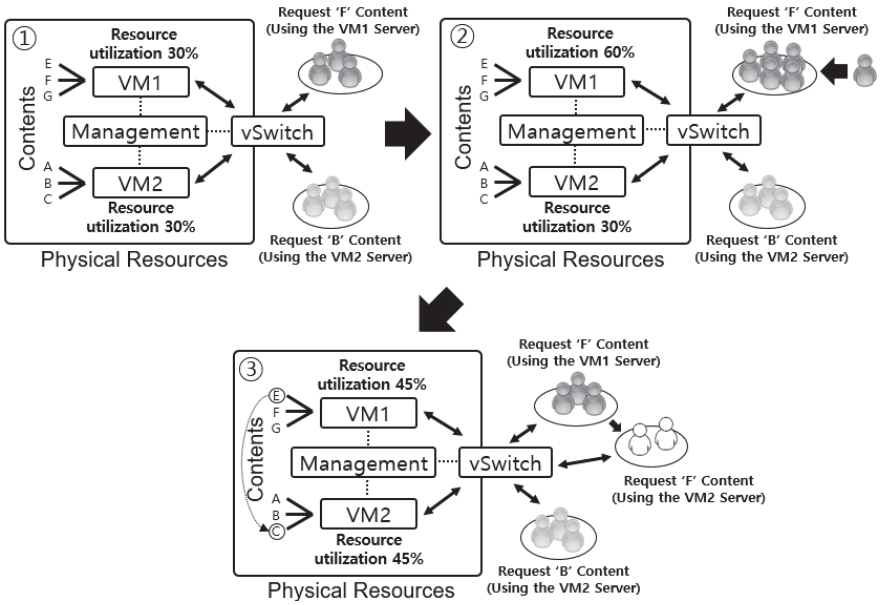

Fig. 2 VM resource utilization depending on the usage scenarios

5 Experimental Results

We want to proceed with the simulation test for the VM resource utilization method according to the division method and server utilization applying the algorithm proposed above. A Lab was developed to test the experiment using a Windows-based program in C# to accept the actual service delivery environment; it was carried out to test the assumption of the time to copy data to support streaming. The data size of 3 GB of the program was used to transfer the image file. The type of test carry out is a total of four types: if the segmentation algorithm is not applied with a single server, when the segmentation algorithm is not applied with a VM server, if the segmentation algorithm is apply using a single server, and finally, segmentation algorithms apply using the VM server. Using the VM creation based on a program in C#, the dynamic experiments were carried out.

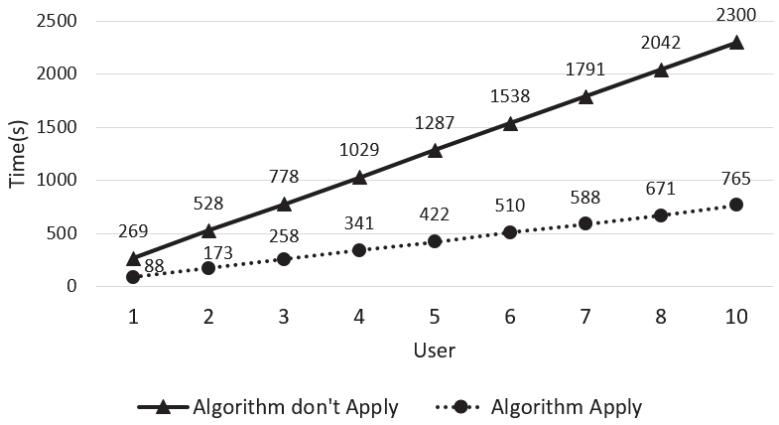

Fig. 3 Comparison results according to the split delivery algorithm.

As shown in Figure 3 in order to provide optimal streaming multimedia services, are when split algorithm is applied and when the algorithm is not apply to the test. At the same time the number of people who request and receive the service, like the situation shown in Figure 1 to test the configuration. If it is not a split stream from the experiments, it did not have any problem in the streaming service in a simple manner when the P2P service received large high quality images is down or hang occurred. Was also in similar time receive the user downloads the time to play the real-time streaming. We have a goal to be able to use the fastest servers around the user when applying the proposed algorithm. Thus, when not applying an algorithm to support a faster data rate it can be confirmed through this test.

6 Conclusion

In this paper, provide optimum multimedia services to users based on the network environment using OpenStack VM. The user can navigate to a fast server to receive multimedia streaming service and provide optimal multimedia services from a server search. However, the absence of the content requested by the user to the fast server is connected to another server to receive the streaming service, at the same time, by transmitting the content to the server is optimized to the user from the other server; the user may be provided with an optimized multimedia service. In order to apply an environment such as this, we split the streaming algorithm to an optimized server to the user by distributing the data quickly to the center proportionally to the synchronization. We have used the proposed algorithm by doing comparative testing and analysis; this has been confirmed that a common non-split transmission method using the P2P transfers. We can transmit data faster than the conventional transmission method by applying to the proposed algorithm. These results show that it is possible to send to the user multimedia data in a sufficient quality. Also we have to configure each server VM by providing a multimedia service to a user, depending on the frequency content of each batch and fast search is possible, with the use of a single physical resource efficiently. In this paper, as shown in Figure 1 and 2 network configuration testing, also a wide variety of environments can be applied for faster multimedia streaming approach will be an ongoing research.

Acknowledgments This research was supported by the IT R&D program of MSIP(Ministry of Science, ICT and Future Planning) / NIPA(National IT Industry Promotion Agency). 12221-14-1001, Next Generation Network Computing Platform Testbed

References

1. Wang, B., Sen, S., Adler, M., Towsley, D.: Optimal proxy cache allocation for efficient streaming media distribution. In: INFOCOM 2002: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1726–1735. IEEE (2002)
2. Almeida, J., Eager, D., Ferris, M., Vernon, M.: Provisioning content distribution networks for streaming media. In: INFOCOM 2002: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1746–1755. IEEE (2002)
3. Eager, D.L., Ferris, M.C., Vernon, M.K.: Optimized caching in systems with heterogeneous client populations. Performance Evaluation **42**(2), 163–185 (2000)
4. Verscheure, O., Venkatramani, C., Frossard, P., Amini, L.: Joint server scheduling and proxy caching for video delivery. Computer Communications **25**(4), 413–423 (2002)
5. Almeida, J., Eager, D., Vernon, M., Wright, S.: Minimizing delivery cost in scalable streaming content distribution systems. IEEE Transactions on Multimedia **6**(2), 356–365 (2004)

6. Lin, B., Dinda, P.A.: Towards scheduling virtual machines based on direct user input. In: Second International Workshop on Virtualization Technology in Distributed Computing, VTDC, p. 6 (2006)
7. Korotaev, K.: Hierarchical cpu schedulers for multiprocessor systems, fair cpu scheduling and processes isolation. In: IEEE International Cluster Computing (2005)
8. Patil, S.S., Bhavani, K.: Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment. International Journal of Engineering and Advanced Technology (IJEAT) **3**(6), 1107–1117 (2014)
9. Yadav, S.: Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, Openstack and Opennebula. International Journal of Engineering and Science **3**(10), 51–54 (2013)