Dividing Transmission Method for Multimedia Service Using OpenStack

Sanghyun Park[1], Linh Van Ma[2], Jinsul Kim[3]
School of Electronics and Computer Engineering, Chonnam National University, 77
Yongbong-ro, Buk-gu, Gwangju 500-757, Republic of Korea
sanghyun079@gmail.com[1], linh.mavan@gmail.com[2], jsworld@jnu.ac.kr[3]

Abstract - *This paper proposes a dividing transmission method for providing an optimal delivery multimedia content using OpenStack. Multimedia data are installed in several servers, these servers has a Management Server which provides an optimal method when user requires content. In addition, we develop a partition algorithm so that users can receive multimedia data from these servers in an appropriated way through Management Server. In this paper, a fast algorithm for getting data in the multimedia server have developed based on the transmission of each server.*

Keywords: OpenStack, Division Transmission, Optima Multimedia Transmission, Management Server

1 Introduction

Various techniques have been researched in order to provide multimedia services globally. In addition, many agencies and suppliers are developing their business along with cloud services so users can require multimedia services at any time over the Internet [1-2]. However, building a real-time multimedia system service is difficult. In addition, to serve large amount request from users, we need manage the system efficiently. If the user requests data from a multimedia server that is far away or it is overloaded, users cannot have a seamless service. Therefore, in this paper, we propose an effective management resources method and a dividing algorithm that provides an optimal multimedia service in OpenStack [3]. Dividing transmission algorithm is a method that transmit data rapidly from a nearest server that contains data to a user [4]. If the server has no desired multimedia contents, it dedicates other servers that provide the requesting data even it has fast connection. In general, the transmitting multimedia data from the server that holds the content cannot provide an optimum if the server is far away from a user. Therefore, this paper uses the optimal selection server method user-oriented-based for the transmission of multimedia content that uses the transport techniques OpenStack. We describe the algorithm for split delivery and test the proposed system as well.

2 Related works

Many studies have been conducted with multimedia services. Among them, the multimedia transmission and related articles "Caching and optimized request routing in cloud-based content delivery systems" paper [5]. It limits storage costs and physical resources with a variety of uploading content and provides user-requested content quickly. It also described a method of how to reduce the delay time. The proposed method in this scheme serves the content using dynamic caching, the Elastic storage resources that are less costly than the existing algorithm. The "Volly Automated Data Placement for Get-Distributed Cloud Services" paper [6], describes an increasing in the size of the distributed data center services, data distribution considerate the costs and limitations of the WAN bandwidth with data center capacity. The proposed method has been described that an optimized algorithm development and connection request access patterns according to the data center and the log acquisition of the client location. In addition, this paper analyzes the log data acquisition using its proposed algorithm, then the data analysis was applied to a cloud service. Figure 1 shows a flowchart of data analysis in the paper.

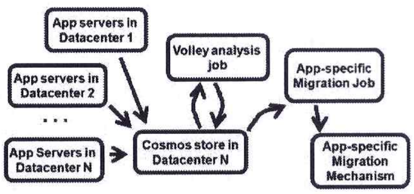

Fig. 1. Data flowchart analysis of Volly automated data placement

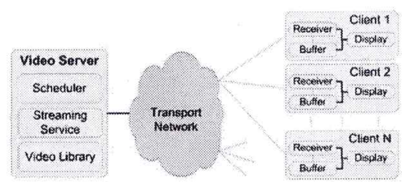

Fig. 2. Structure of video delivery network

"In the paper "Improving Scalability of VoD Systems by Optimal Exploitation of Storage and Multicast" stated that VoD system users and an amount of data increases continually in accordance with the scalability and bandwidth VoD services. Therefore, in order to solve the above problem, Prepopulation assisted batching with multicast patching (PAB-MP) developed for End User Devices to read an initial segment of the pre-reading and video multicast server with fast and simple structure. Figure 2 shows the structure of the network proposed in the paper. Multimedia transmission method of the above paper is that no matter how good they are based on the server, the server does not provide built even useless to users. In this paper, we developed an algorithm that provides multimedia content then selected the optimum server to incorporate each one in the center server for the user and proceed an experiment.

3 Operating structure of whole system

3.1 Scenario of the whole system

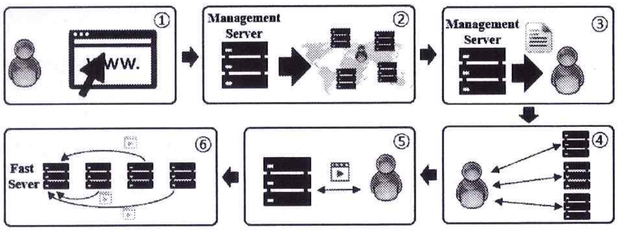

Fig. 3. Proposal system sequences

The overall structure of the system is shown in Figure 3. Fig 3-① user first uses the Web and requests the server for selecting the desired content. Fig 3-② OpenStack-based Management Server retrieves the servers which near the user's location. Fig 3-③, the Management Server sent the user the list of the adjacent VM Data Servers. Fig 3-④, users request the respective data by using the list of VM DataServer which received from the Management Server, then the servers are requested to check a transmission rate of the user. Fig 3-⑤ a user check the transfer rate of the server, and then select the server volume. And it receives the multimedia data transmission request from the selected server. Fig 3-⑥ If the server does not provide optimum content transmission which requested by the user in the VM DataServer, user requests a file from the other server. At this time, the user transmits data that divided proportionally according to the data rate at which the file is requested to provide a respective server in multiple servers. When the split delivery to the user, the best portions of the fast server sends to the user multimedia content, it also receives the content from some other server. By this way, the user requests the content easily using the Web and the internal server system provides a good quality content to the user. We use a server-centric rather than user-centric data transfer with data transmission faster than any existing data server. In general, the service rate of the server cannot provide a seamless multimedia content due to the switch, router, and relay device that connects them from the remote server. Therefore, if user's surrounding servers provide a data center to the user may send faster and better quality multimedia data than the existing data servers.

3.2 Division transmission method of real-time multimedia

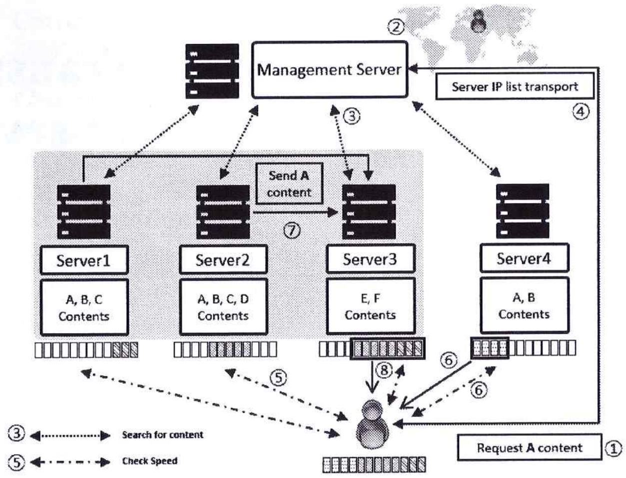

Fig. 4. Data division for optimal multimedia scenario transmission

Figure 4 shows the data division transfer scenarios for optimum multimedia transmission. Fig 4-① users use a variety of devices with Web access and select multimedia content. Fig 4-②, ③ After analyzing the location of the user device by using the IP, the server searches for the nearby multimedia content providing servers. Then, it checks the status of the multimedia content that requested by the user with the retrieved destination server. Fig 4-④ Management Server transfers the list of discovered servers to users. Fig 4-⑤ The user checks the response speed of the respective server, using the list received from Management Server. Fig 4-⑥ user checks the response rate of the data piece on server in order to request the content from the second fastest if there is no content in the second fastest Server4. The second fastest Server4 sent real-time multimedia content to users earlier than the rest of the content that requested by the user from the other, such as Fig 4-⑦ are partially transfer the content from the fastest Server3. At this point, the partial rate based on the amount of transmitted data servers that typically divides the multimedia data. In this way, it can transmit multimedia data of high resolution and high quality. Finally, in Fig 4-⑧ if the content transfer is completed with fastest connections and terminate at the server (Server 3) then data is merged from data piece in the second fastest server (Server4) and the fastest server (Server 3). This is a user-oriented optimum method for transferring data in between servers and user, the user finally has a high resolution and high-quality transmission of multimedia data.

3.3 Dividing transmission algorithm for optimal transmission of multimedia content

Algorithm 1 is divided transmission algorithm for optimized transmission of multimedia content. $S_c(i)$ refers to a server that has the content requested by the user. After the server is ready to transmit data, the server finds in Management Server, the fastest server in the list Sort(0) which will transmit data from the fastest optimum server $S_c(0)$. The server will provide the requested content to the user, if the server has different content then a next fastest server is used to provide data to users. The remaining servers are the servers that provide the optimal rate content on the size of the user-requested content, which is transmitted to the user proportionally transmitted to the server list Sort(0). At this point, the expression (1) means that the entire file, and divided according to the percentage of file size S_f. $S_f(0)$is the first front part of the multimedia data to be transferred. $S_b(i+n)$ is sent directly to the user, except for servers that belong to the servers $S_c(i)$ based on the speed ratio S_b. The transferring data in server $S_c(0)$ ends or the best Sort(0) server transfer data is completed, then the user receives the provided requested-content. Division transmission algorithm is shown in the Fig. 4-④ ~ ⑥.

Algorithm 1. Division transmission algorithms for multimedia transferring content

WHEN $\boldsymbol{S_c(i)}$ Server ready for data transmission **THEN**
  IF $\boldsymbol{S_c(0)}$ **! = Sort(0) THEN**
  // If you do not have the fastest server Content
    $\boldsymbol{S_c(0).send();}$
    // Earlier data transmission of the multimedia data
    WHILE $\boldsymbol{S_c(i+1)}$!= NULL **THEN**
    // Data partitioning according to server ratio

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

      $\boldsymbol{S_c(i+1).send();}$
      // Data transmission according to the ratio
    END WHILE
  END IF
  IF $\boldsymbol{S_c(0)}$ **== Sort(0) THEN**
  // If you have the fastest server content
    $\boldsymbol{S_c(0).send();}$
  END IF
  IF (Complete $\boldsymbol{S_c(0)}$ data transfer) or (Complete Sort(0) data transfer) **THEN**
    Sort(0).Send();
    // If you have the fastest server content
  END IF
END WHEN

4 Division transmission experiment and evaluation result of multimedia content

In this paper, the algorithm was developed by dividing the actual data transfer. It is possible to use the simulation tools for testing, but we would rather develop a test program to find out how the rate comes out in actual application test. Fig. 5 shows the user is receiving data from the test program. Data Management Server uses a Linux-based OpenStack environment, the user client program test was developed using a Windows-based environment of the C# language. When a user connects to the server as shown in the figure, it shows a content list from the Management Server. A user chooses content by clicking on the content list which receives from the server that holds on the Management Server. The testing programs divided multimedia data transmission proportionally.

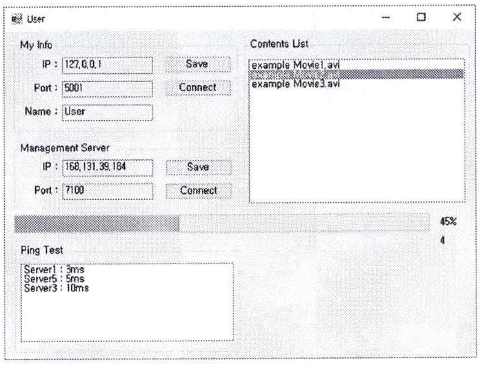

Fig. 5. Test program for division transmission speed

In order to receive offers from servers, we must measure the speed of servers. In this paper, we use ping testing method. First, the test-program sorts servers in the order which is proportional to the speed and transmits the data according to the split command. Secondly, the user requests the multimedia data from the data server. Third, multimedia data is sent to the user after the playback, it sends all the rest of the data files. Finally, the combined data is provided to the user. As such, the user receives the transferred data is partitioned on the basis of the fastest server in order, which is shown in sort expressions (2) with share proportionally data.

$$\frac{S(i)_{speed}}{Sum_{S\_speed}} F_{size} = S(i)_{F\_size} \qquad (2)$$

$S(i)_{Speed}$ refers to the speed of each of the data server, $Sum_{S_speed}$is a server that provides data to users throughout the speed, $F_{size}$is the size of the data requested by the user, $S(i)_{F\_size}$ refers to the amount of data to receive data transmitted from the server. For example, if the total file size of 360MB in server 1 then the partition is allocated about 130MB. If we divide the capacity of data four times in the server, we get 32MB data allocation. If the data is allocated to the data server informs the start and end points. The data server uses the start and end points if the pieces of data are sent to the user. The first part of data is sent from the fastest server to user meanwhile user also receives other pieces of data from the other servers.

Table 1. Experiments environment network condition

	Server1	Server2	Server3	Server4	Server5
Providing speed	6Mbps	3Mbps	2Mbps	1.5Mbps	4Mbps

Fig. 6. Partitioning multimedia data

Fig. 7. Data paritioning and its combination

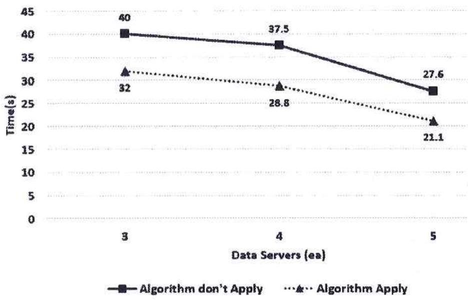

Fig. 8. Division transmission test result of multimedia data

Fig. 6 shows a picture of reproducing the first data piece. It is possible to play the video partially as shown in the rectangle because it transmits the data into small pieces. Then, when part of the data transfer is complete, the data is combined into a single file can reproduce the entire multimedia file. Figure 7 shows the three received data pieces which deliver from multimedia data servers with 1-3.avi, and the combined multimedia data file is Movie2.avi. Fig. 8 shows the results of the speed test multimedia data transferred with the graph partitioning. The vertical axis represents the amount of time (in seconds), the horizontal axis represents the number of the server. Because the data is partitioned with rate 1:1 as shown in the image which makes the transmission typically faster, when we apply the algorithm to split data into the proportional piece, then we achieve transfer data faster than keeping the original file. The above program uses an algorithm in the real to measure whether the tests and data transfer, we might have some errors depending on the speed and the surrounding environment. Therefore, we use the extension number of the same data and use the server to get the accurate measurement under the same conditions, and measure the transmission rate.

5 Conclusions

We proposed a method for the user to find the optimal server center dividing multimedia content delivery in order to provide optimal multimedia services. In the experiment in this paper, we divide the transmission data based on a rate from Ping Test, The division number of the receiving server through the experimental results confirm that the speed of the content to be provided is reduced. In order to provide a service we use OpenStack, OpenStack only uses for server environments, In order to evaluate the proposed method, we developed the program on C#-based. In the future, we will continue the research for delivery data from dynamic VM environment resources efficiently.

6 Acknowledgements

This research was partially 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] and, also supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2013R1A1A2013740)

7 References

[1] Jieyao Liu, Ejaz Ahmed, Muhammad Shiraz, Abdullah Gani, Rajkumar Buyya, and Ahsan Qureshi, "Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions," Journal of Network and Computer Applications, Vol.48., pp.99-117, 2015.

[2] Nan, Xiaoming, Yifeng He, and Ling Guan, "Towards optimal resource allocation for differentiated multimedia services in cloud computing environment," Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, pp.684-688, 2014.

[3] Yanagawa, Toshihide, "OpenStack-based Next-generation Cloud Resource Management," Fujitsu Sci. Tech. J, Vol.51 No.2., pp.62-65, 2015.

[4] Ubarhande, Vrushali, Alina-Madalina Popescu, and Horacio Gonzalez-Velez, "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments," Complex, Intelligent, and Software Intensive Systems (CISIS), 2015 Ninth International Conference on. IEEE, pp.217-224, 2015.

[5] Niklas Carlssona, Derek Eager, Ajay Gopinathana and Zongpeng Li, "Caching and optimized request routing in cloud-based content delivery systems," Performance Evaluation, Vol.79., pp.38-55, 2014.

[6] Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu and Alec Wolman, “Volly: Automated Data Placement for Get-Distributed Cloud Services," Networked Systems Design and Implementation, pp.17-32, 2010.