Real-Time Transcoding and Cloud Resource Allocation of Adaptive Multimedia Stream System for Multiple Channels and Multiple Users

Hui-Kai Su[a,], Cheng-Shong Wu[b], Chang-Ming Lee[b], Yun-Ying Zeng[b], Yan-Sun Chu[b], Ching-Lung Chang[c]

[a] Department of Electrical Engineering, National Formosa University, Taiwan (R.O.C.)
E-mail address: hksu@nfu.edu.tw

[b] Department of Communications Engineering, National Chung Cheng University, Taiwan (R.O.C.)

[c] Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan (R.O.C.)

Abstract

With the development in Internet technology, mobile and handheld devices with multimedia compression technologies, the users can surf the Internet to watch video entertainment contents at anytime and anywhere. That is making multimedia streaming service has become an integral part of daily life. Therefore, various internet video services are increasingly completed. In order to provide a suitable platform for heterogeneous video streaming service, it has to transcode on demand. However, real-time video transcoding requires a lot of cloud computing resources, it is difficult to achieve in the general terminal device. With the cloud computing technologies, the computing server in the transcoding cloud will serve and satisfy the requirements of different terminal equipment. This paper proposes a real-time transcoding of adaptive multimedia stream system. To meet the requirement of real-time transcoding, the cloud resource allocation mechanism was also design in this paper. According to the resolution, quality and other conditions desired by the users, the cloud management configures the optimum number of virtual machines to deal with the cloud transcoding task. Finally, the system prototype was design and implemented. In our testbed, two clients were playing live streams with difference programs simultaneously. While the state of available bandwidth is changed, both clients can adapt the stream to fit the network condition. Both video programs were still playing smoothly and continuously. The adaptation delay is about 15 seconds.

Keywords: video transcode, cloud computing, resource allocation, adaptive multimedia stream

1. Background/ Objectives and Goals

With the development in Internet technologies, mobile and handheld devices with multimedia compression technologies, the users can surf the Internet to watch video entertainment contents at anytime and anywhere. That is making multimedia streaming service has become an integral part of daily life. Therefore, various internet video services are increasingly completed. In order to provide a suitable platform for heterogeneous video streaming service, it has to transcode on demand. However, real-time video transcoding requires a lot of cloud computing resources, it is difficult to achieve in the general terminal device. With the cloud computing technologies, the computing server in the transcoding cloud will serve and satisfy the requirements of different terminal equipment.

Moreover, to transmit multimedia as much as possible, User Datagram Protocol (UDP) is frequently adopted in network. However, UDP is unreliable because of lacks of congestion control mechanism. Unfortunately, the fast growing real-time application may lead Internet into congestion collapse, especially with the unresponsive UDP protocol over the error-prone wireless network. In opposition, the transmission rate variation is considered by Transmission Control Protocol (TCP). According to the network transmission quality, the transmission rate and request of retransmission of lost packets can be adjusted by the Additive-Increase/Multiplicative-Decrease (AIMD) **Y. R. Yang & S. S. Lam (2000)** which realizes the congestion control. Consequently, the transmission of multimedia streaming can be achieved completely. The perception of quality change remains, because the conditions (buffering, playback time, etc.) in the client part are rarely regarded in the congestion control.

Recently, HTTP Live Streaming (HLS) has been released by APPLE for HTTP-based media streaming communications protocol. However, there are several limitations, like company-independent streaming servers as well as playback clients. Based on TCP, pull-based media streaming protocol can check URL information in the segment list received by the client and transmit segments with competent quality by HTTP over TCP. This scheme can support the compatibility among various platforms and completeness of media transmission in the open network. In this trend, Moving Picture Experts Group (MPEG) developed MPEG-DASH (Dynamic Adaptive Streaming over HTTP) in 2012 **I. Sodagar (2011)** to standardize the compatibility among various devices or servers. In this protocol, the packet format, segmentation of steaming file, and segment delivering are defined to provide the flexibility to design an adaptive algorithm for the dynamic network environment. To smooth HTTP throughput, several adaption schemes based on bandwidth conditions **C. Liu, I. Bouazizi & M. Gabbouj (2011)** and buffer conditions **L. De Cicco & S. Mascolo & P. Vittorio (2011)** and **C. Zhou & C.-W. Lin & X. Zhang & Z. Guo (2014)** instead of instantaneous TCP transmission rate are proposed to improve the performance of the HTTP

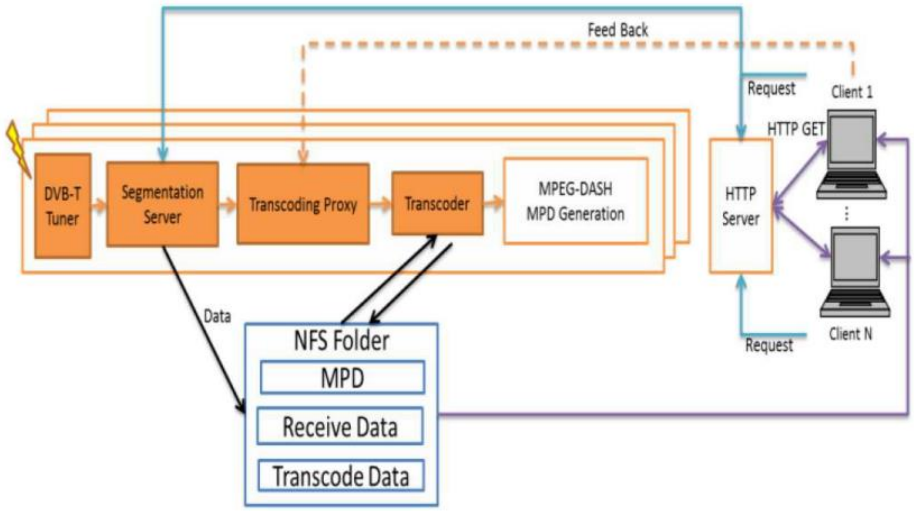

Fig. 1: The system architecture.

streaming services. The challenge remains on the combination of these issues to smoothly adjust the video bitrate or the quality of experience (QOE) over the video streaming services.

The goal of this paper is to design a cloud transcoding system to achieve real-time transcoding for live streams. While the state of available bandwidth is changed, the video should be adapted to a suitable video quality and meet the real-time requirements of live streams.

2. Methods

2.1 System Architecture

This paper proposes a real-time transcoding of adaptive multimedia stream system, which is shown in Fig. 1. Based on the IaaS (Infrastructure as a Service) cloud transcoding technologies, the live programs of multi-channel DVB-T will be transcoded for various devices, such as HDTV (High Definition Television), personal computer, smart phone, etc. Moreover, the transcoded streams are constructed with MPEG-Dash format. When the network state changes, the terminal devices will trigger to upgrade or downgrade the video quality by monitoring the state of receive buffer. Sequentially, our system will transcode the suitable video quality for the terminals on demand. Thus, the terminal devices can receive the adaptive stream with different qualities according the available bandwidth in transportation networks.

2.2 Cloud Transcoder

To realize a real-time video transcoding service, an IaaS system is constructed as a transcoding cloud. Fig. 3 presents the components of the proposed cloud system. Font-end is in charge of managements in the transcoding cloud, including ONED, drivers, guest OS images, monitoring of system resource in physic and virtual machine, etc. Cluster node can offer the computation resource by dynamically processing and removing the VM instance which contains an individual guest OS and provides the transcoding service. Transcoding proxy (TP) can provide the NFS storage service and process the dynamic allocation (proposed in Section III-C. Therefore, each VM can access the corresponding video chunks before transcoding. MPEG-DASH formatting can packetize the transcoded chunks and generate MPD for the client.

2.3 Earliest finishing time first VM allocation

To meet the requirement of real-time transcoding, the cloud resource allocation mechanism was design in this paper. Some transcoding task just need one VM (virtual machine), but some complex transcoding task requires more computing resource with multiple VMs. According to the resolution, quality and other conditions desired by the users, the cloud management configures the optimum number of virtual machines to deal with the cloud transcoding task. Thus, the earliest finishing time first VM allocation was proposed in this paper.

When TP accesses a video chunk in the cloud, this allocation mechanism can estimate the transcoding time in each VM and select the fast one to realize the transcoding of this chunk. The dynamic update of transcoding rate for each VM is necessary to successfully estimate the transcoding time and adjust the number of virtual transcoder to satisfy the required transcoding delay. The transcoding rate is normalized by dividing the playout time of a chunk (several segments) by the transcoding time. Several parameters are defined as follows.

τ_j^i : Processing time of ith chunk in jth transcoder.

t^i : Arrival time of ith chunk.

$\boldsymbol{T^i}$: Playout time of ith chunk.

f_j^i : Finish time of ith chunk on jth transcoder.

$D_{\max}$: Maximal transcoding delay.

R_j : Transcoding rate of jth transcoder.

The transcoding rate depends on the desired quality, video characteristics, capability of VM, and payload of physical machine. The policy is described below.

Step 1: Find the initial number of available VM by

$$K = \arg\left(\min_n \left(\sum_{j=1}^{n} R_j > 1 \right) \right) \tag{1}$$

Step 2: Estimate the ith chunk transcoded by the jth transcoder as

$$\boldsymbol{\tau}_j^i = \frac{\boldsymbol{T}_i}{\boldsymbol{R}_j} \tag{2}$$

Step 3: Update the transcoding rate in jth transcoder when a chunk is transcoded by the jth transcoder with time τ.

$$R_j = (1-\alpha)R_j + \alpha\frac{T}{\tau} \tag{3}$$

Step 4: In time t^i, estimate the transcoding time of ith chunk in the jth transcoder with pending or workload as

$$f_j^i = \max(t^i, f_j) + \tau_j^i \tag{4}$$

Step 5: Choose the fast one

$$\hat{\boldsymbol{j}} = \operatorname*{argmin}_{j} \boldsymbol{f}_j^i \tag{5}$$

and estimate the ith chunk transcoded by the $\hat{j}$ th transcoder by

$$f_{\hat{j}}^i = f_*^i \tag{6}$$

Step 6: Increase the number of transcoder while

$$f_j^i - t^i > D_{\max} \tag{7}$$

holds for the original maximal delay $D_{\max}$ in the Front-end.

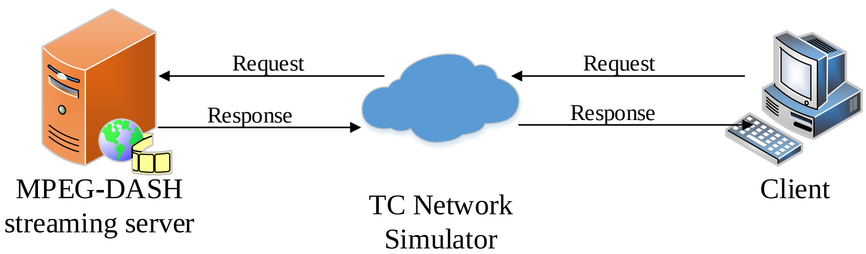

Fig. 2: The paradigm of test environment.

3. Results

The test environment in this paper is shown in Fig.5. MPEG-DASH video streaming data is transmitted between HTTP server and client with unicast (HTTP over TCP). The Traffic Control (TC) is realized by the Linux kernel in the HTTP server. The resolution of original video streaming is 1920x1080 with bitrate 13 Mbps and framerate 30 Hz. Three qualities considered in this transcoding cloud are Q_{low} for 1Mbps, Q_{mid} for 3 Mbps, and Q_{high} for 5 Mbps. The performance of proposed algorithm is evaluated with the available bandwidth variation. The initial delay is about 16 seconds. The test environment is illustrated in Fig. 2.

In this experiment of cloud resource allocation, round Robin (RR) algorithm and Most Powerful Available Transcoder First (MPATF) algorithm are considered for comparison that are shown in Table I. With RR, the sequential chunks are allocated in the corresponding VMs in order by the transcoding proxy. The number of transcoders would be increased while the assigned VM is busy (not available). Instead of sequential assignment, MPATF can select the most powerful VM in the pool (containing all available VMs) to finish the transcoding as fast as possible. In the result, we can observe that our proposed scheme (EFTM) can provide real-time transcoding with the minimal number of transcoders.

Table 1: Resource consumption in the transcoding cloud.

Utilization	TC1	TC2	TC3	TC4	TC5	Number of transcoder
RR	52.12	51.68	52.31	50.36	44.77	4.46
MPATF	63.08	64.97	18.96	2.96		2.96
EFTM	81.91	81.04	45.64			2.1

For the adaptive streaming, in our testbed, two clients were playing live streams with difference programs simultaneously. Moreover, the available bandwidth of each client is controlled by the WAN emulator that was achieved by “tc” command in Linux platform. Figure 3 and Figure 4 are shown the test results. While the state of available bandwidth is changed, both clients can adapt the stream to fit the network condition. Both video programs were still playing smoothly and continuously. The adaptation delay is about 15 seconds.

4. Rerefence

Y. R. Yang & S. S. Lam (2000). General AIMD congestion control. In Proc. of International Conference on Network Protocols, 187–198.

I. Sodagar (2011). The MPEG-DASH Standard for Multimedia Streaming Over the Internet. IEEE Multimedia, 62-67.

C. Liu, I. Bouazizi & M. Gabbouj (2011). Rate Adaptation for Adaptive HTTP Streaming. In Proc. ACM MMSys11, 169–174.

L. De Cicco & S. Mascolo & P. Vittorio (2011). Feedback control for adaptive live video

streaming. In Proc. ACM MMSys11, 145–156.
C. Zhou & C.-W. Lin & X. Zhang & Z. Guo (2014). A Control-Theoretic Approach to Rate Adaption for DASH Over Multiple Content Distribution Servers. IEEE Trans. Circuits. Syst. Video Technol., 24(4), 681–694.

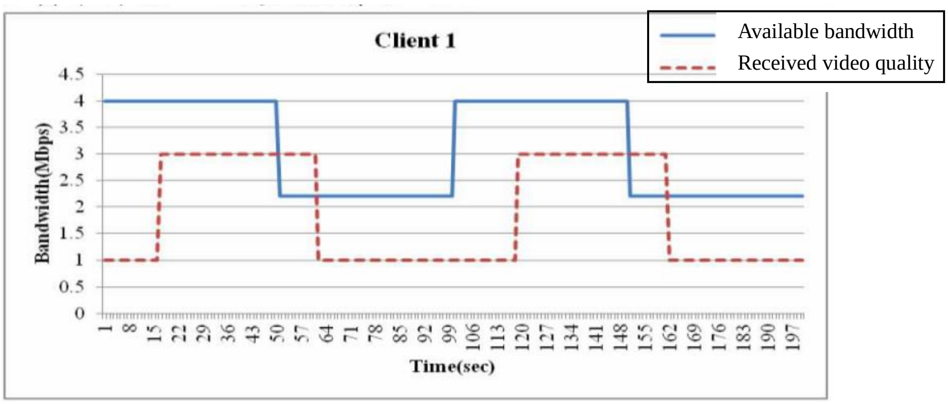

Fig. 3: The test result of client 1.

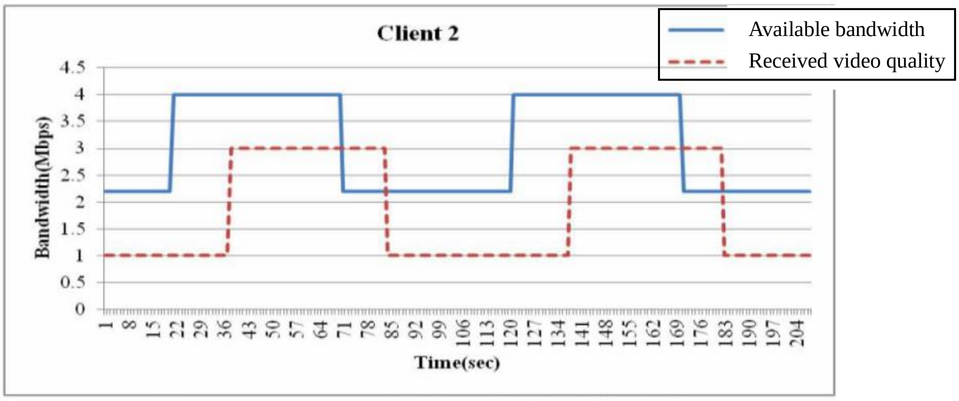

Fig. 4: The test result of client 2.