미디어 클라우드에서의 적응형 스트리밍을위한 최적화 된 멀티미디어 트랜스 코딩 방법

Linh Van Ma[*] · 박상현[**] · 김진술[***]

[*] [**] [***] 전남대학교 전자컴퓨터공학부

An Optimizing Multimedia Transcoding Method for Adaptive Streaming in Media Cloud

Linh Van Ma[*] · Sanghyun Park[**] · Jinsul Kim[***]

[*] [**] [***] School of Electronics and Computer Engineering

Chonnam National University

E-mail : linh.mavan@gmail.com, sanghyun079@gmail.com, jsworld@chonnam.ac.kr

Abstract

In this research, we address the issues of diversity requests from users in a cloud streaming environment, by modeling the problem in the form of an optimal problem. We use the moving average to deal with server state fluctuations. We then find an optimal server for a given request coming from a client. In the experiment, we implement the optimization method with DASH streaming to demonstrate the effectiveness of our optimized solution. We create several servers running on the Docker platform to simulate the cloud environment. Experimental results show that our methodology reduces the time of the transcoding process up to 30% compared to existing research.

Keywords

Cloud Computing, Adaptive Streaming, Transcoding, Data Replication, Docker

Ⅰ. Introduction

Adaptive streaming is a video streaming technology where video quality can vary depending on the state of the network and the processing capabilities of devices. Recent studies have not paid much attention to the choice of data centers to serve user requests in the adaptive streaming. They only interested in standard parameters when choosing a serving server, such as bandwidth or server load, or a combination of both. Furthermore, these studies only consider the present value of cloud measurement parameters, such as server load, network speed. These research methods will sometimes result in rough estimates or non-optimal options. So in this article, we propose an optimization method to ensure the quality of video streaming in Dynamic Adaptive Streaming over Hypertext Transfer Protocol (HTTP) or so-called DASH in a cloud environment. Specifically, by using a server that manages other data servers in the cloud, we redirect user requests to data servers, to optimize the performance of the network and adapt to users' requests.

The paper is organized as the following. Section 2 discusses some related research, topics in adaptive streaming. In Section 3, we introduce and model the problem. In section 4, we implement a cloud system using Docker [2] and Node.js. Finally, we conclude our paper and propose future research of the approach.

Ⅱ. Related Research

In Adaptive Streaming, transcoding multimedia to different bitrates and qualities requires extensive computation resource. It also occupies the most time in the stages of delivering video streaming to a user. Therefore, we need a cloud environment that can handle such high computing demands. Similar to our approach to reducing the time of transcoding, the authors in [1] proposed a video transcoding scheduling method for DASH in the cloud. They prioritized each job of transcoding and adjusted transcoding mode based on the cloud system load. In that way, they reduced video transcoding completion time and balanced the cloud system load as well as smoothing video playback at a client side.

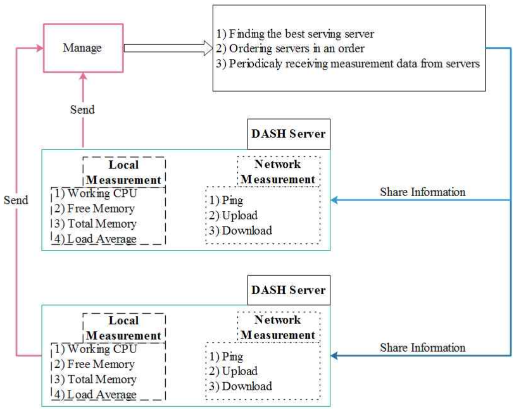

Fig. 1. Management server manages parameters from other servers.

Ⅲ. System Overview

In this article, we use seven parameters to evaluate the performance of a DASH streaming server. Regarding local measurement, we have; 1) Working CPU, 2) Free memory, 3) Total memory, 4) Load average. Regarding network measurement, we have; 5) Ping, 6) Upload speed, 7) Download speed. As shown in Fig. 1, management server periodically receives measurement information from DASH servers. The management server finds a server that best serves a request from a client. It also orders the DASH servers into an order. The management server shares information about the order of the servers for all servers through a synchronization mechanism using a file-sharing protocol such as File Transfer Protocol (FTP). DASH servers use this information for the purpose of retrieving DASH files from other servers in case it cannot serve a transcoding request from a client. This technique is a solution for sharing DASH files of a server when it contains a multimedia video while other servers do not have the video content.

In the evolution metrics, we consider two factors on a client side to evaluate the response time of a cloud system to a user. First, start latency describes the time interval between the time when a client uploads a video source to the cloud and the time when the video is available for playback. Second, the number of representations represents a quality range which the cloud can provide for a user. Our goal is to minimize the total start latency with the limited of cloud resource and network throughput. Reasonably, the time of the transcoding process increases if server load increases and the network bandwidth decrease. We use the moving average in a short period to reduce the fluctuations of measurement values retrieving from a server.

Ⅳ. Cloud DASH Streaming Experiment

In this section, we simulated a cloud system using Docker. We installed a DASH streaming system and made a Docker container which contains the DASH server as well as runtime environment. The DASH implementation is Node.js-based system. We installed several modules to support adaptive streaming in the DASH streaming. We used Fast Forward Motion Picture Experts Group (FFMPEG) software to transcode a multimedia video file into different representations. We used MP4Box software to generate a Media Presentation Description (MPD) file which contains information of DASH streaming information. We used the dash.js open source to build a player client, which can change video representations along with fluctuation of throughput. We installed our proposed optimization method in the management server, where we consider all requests from a client to solve the optimization problem.

Table 1. List of video source file in the experiment.

Source Video	Max Resolution	Duration
4K Hawaii Drone Footage	2160p	11m10s
NewYork_In_4K	2160p	4m41s
SunflowerPaddy Sun	480p	3m32
Sunflower - Paddy Sun	360p	3m30s
[SIXTEEN] Happy	720p	2m31s
TWICE_TT	720p	4m13s

We tested with six video source files obtained from YouTube as shown in Table 1. We implemented a program in servers, which transcodes a source video from 144p to the maximum vertical resolution of the video. In that way, we ensure that users can watch streaming video in different qualities depending on the quality of the network.

We can run the Docker container directly without installing any addition libraries or runtime environment. In the method illustration, we compared the proposed method with First In First Out (FIFO) scheduling policy in a cloud environment [3]. We ran ten DASH streaming server in which one server is a management server, and others act as data servers. As a result shown in Fig. 2, our method reduces transcoding time around 30% compared to the FIFO methods.

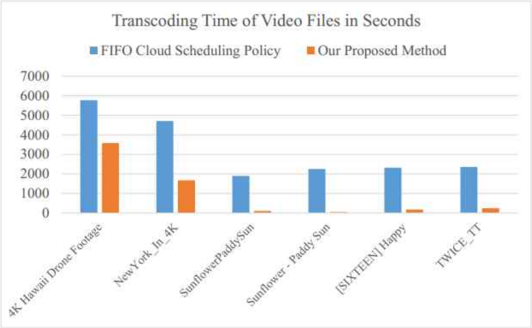

Fig. 2. Multimedia transcoding time comparison.

Ⅴ. Conclusion

In this article, we found an optimal server for a given request coming from a client. We proposed a method which reduced the time of the transcoding process in DASH streaming. As a result, our approach reduced the transcoding time up to 30% compared to existing research. In the future research, we will expand the optimization problem by restricting its conditions corresponding to the cloud resource limitation. It will lead us to the best optimal solution to use the media cloud more efficient.

Acknowledgement

This work was supported by 'The Cross-Ministry Giga KOREA Project' grant from the Ministry of Science, ICT and Future Planning, Rep. of Korea(GK16P0100, Development of Tele Experience Service SW Platform based on Giga Media). Also, this research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] Ma H, Seo B, Zimmermann R, "Dynamic scheduling on video transcoding for MPEG DASH in the cloud environment", Proceedings of the 5th ACM Multimedia Systems Conference, pp. 283-294, 2014.03.

[2] Negus C, "Docker Containers (includes Content Update Program): Build and Deploy with Kubernetes, Flannel, Cockpit, and Atomic", Prentice Hall Press; 2015.12.

[3] Li J, Ma T, Tang M, Shen W, Jin Y, "Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing", Information, Vol.8 No.1 pp.1~25, 2017.02.