Cloud-Based Video Streaming for Energy- and Compute-Limited Thin Clients

Xiangbo Li[1], Mohsen Amini Salehi[2], Magdy Bayoumi[1]
1 The Center of Advanced Computer Studies (CACS)
2 Computer Science Department
University of Louisiana Lafayette
{xxl8948,amini,mab}@cacs.louisiana.edu

Introduction

According to Global Internet Phenomena Report [1], video streaming constitutes about 78% of all the U.S. Internet traffic and this is expected to increase to 80% by 2019.

Video contents, either in form of on-demand streaming (e.g., YouTube [2]) or live-streaming (e.g., Livestream [3]), need to be transcoded (i.e., converted) based on the supported video formats, resolution of the client devices. Since video transcoding is a computationally expensive operation,it is commonly carried out on the client end. Nonetheless, video transcoding consumes a lot of computational power and drains energy which is particularly problematicin the compute and energy-limited devices, such as smartphones, tablets, and laptops.

One approach to address this problem is to store all variations of the same video on the provider servers. However, given the explosive growth of videos on providers and the large diversity of the client devices' characteristics, this solution remains unachievable. Another approach that we propose is to utilize Cloud providers for video transcoding. In this approach, upon requesting a video, Cloud servers receive the video stream and transcode it (in terms of bit-rate, spatial and temporal resolution) based on the client's device characteristics, such as supported video formats and device resolution.

The challenge in utilizing Cloud providers for video transcoding is how to deploy Cloud resources in a cost-efficient manner. It is needless to say that transcoding of video streams has strict Quality of Service (QoS) demands (e.g., deadline or minimum startup delay) that has to be respected. Therefore, the challenge is how to allocate Cloud resources so that the minimum cost is incurred without violating clients' QoS demands? The QoS expectations, in particular, are processing video streams within a deadline (i.e., before their presentation times) and minimizing start up delay for the requested video streams.

To address this challenge, in this research, we present a system that allocatesvideo streams to cloud servers(VMs) with the goal of minimizing the incurred cost andminimizing video streams' deadline miss rate and startup delay.

Proposed Approach

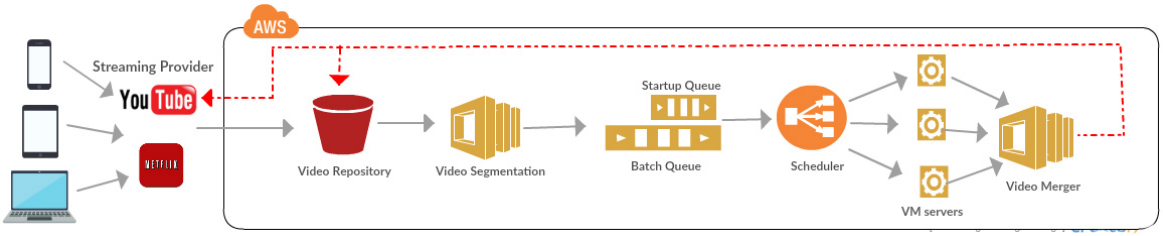

Figure 1. System Architecture

The whole system architecture is shown in figure 1. On the Cloud resources, each video stream received from the video provider, are split into Group of Pictures (aka GOPs), that can be transcoded independently. Each GOP is assigned an individual deadline that is obtained from the presentation time of the GOP in the whole video stream.

The video stream GOPs are batched in a queue upon arrival. To minimize video streams' startup delay, another queue, termed *startup queue*, is also considered. GOPs belong to a new video stream are placed in the startup queue that has a higher priority in compare to the batch queue of GOPs.

A scheduler is responsible for choosing a GOP either from the startup queue or the batch queue and dispatching them to cloud resources (i.e., VMs). The scheduler maps GOPs from the batch queue to the VM with the lowest completion time. The scheduler also prioritizes startup GOPs in the startup queue if and only if they do not cause missing the deadlines of GOPs in the batch queue.

Cloud VMs perform the transcoding operation based on the client's device characteristics. Upon transcoding completion, a video merging module merges the transcoded GOPs to a video stream and sends it to the client device.

Experimental Results

We consider characteristics of VMs in Amazon EC2 [4] Cloud provider for our simulation study. Historic execution times of the GOPs of each video are obtained from the average execution times of the GOPs on Amazon EC2 Cloud. We also used CloudSim [5] discrete event simulator to model our system and evaluate its performance.

Figure 2, demonstrates how the videos average startup delay is reduced when our approach is applied in compare with situation that there is no such policy in place. We observe that, using a startup queue can keep the average startup delay lower than 1 second. The startup delay remains almost the same as the number of transcoded video streams is increased. It's worth mentioning that the reduced startup delay is obtained without major impact on the video streams' deadline miss rate, they all remain ≈ 0%.

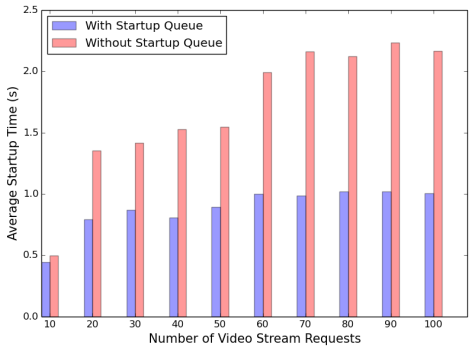

Figure 2. Comparison of average startup time

Conclusion

Cloud-based stream processing enables lightweight video streaming on thin-clients with energy- and compute-limited resources. We proposed a system that reduces the startup delay of video streams without a major impact on other missing the deadline of other video streams. The impact of this project is to increase the number of mobile clients for video streaming providers. More importantly, this project will enable small and medium size providers to increase their clients without major extension on their infrastructure.

Reference

[1] https://www.sandvine.com/trends/global-internet-phenomena/
[2] https://www.youtube.com
[3] https://livestream.com
[4] https://aws.amazon.com/ec2
[5] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and Experience* 41.1 (2011): 23-50.