Resource Allocation for Video Transcoding in the Multimedia Cloud

Sampa Sahoo, Ipsita Parida, Sambit Kumar Mishra,
Bibhdatta Sahoo, and Ashok Kumar Turuk

National Institute of Technology, Rourkela
{sampaa2004,ipsitaparida07,skmishra.nitrkl,
bibhudatta.sahoo,akturuk}@gmail.com

Abstract. Video content providers like YouTube, Netflix cater their content, i.e. news, shows, on the web which is accessible anytime anywhere. The multi-screens like TVs, smartphones, laptops created a demand to transcode the video into the appropriate video specification ensuring different Quality of Services (QoS) such as delay. Transcoding a large, high-definition video requires a lot of time, computation. The cloud transcoding solution allows video service providers to overcome the above difficulties through the pay-as-you-use scheme, with the assurance of providing online support to handle unpredictable demands. This paper presents a cost-efficient cloud-based transcoding framework and algorithm (CVS) for streaming service providers. The dynamic resource provisioning policy used in framework finds the number of virtual machines required for a particular set of video streams. Simulation results based on YouTube dataset show that the CVS algorithm performs better compared to FCFS scheme.

Keywords: Cloud, Video transcoding, Merger, Video Stream, QoS

1 Introduction

Now the content viewing has shifted from traditional TV system to video streaming in laptops, smartphones, etc. through the Internet. Content providers like YouTube, Netflix, Hulu as well as TV channels cater their content, i.e. news, shows, live events, user generated content on the web which is accessible anytime anywhere. According to Cisco Visual Networking Index (VNI) report published in the year 2016, video streaming will escalate up to 82% of the total network traffic by 2020, up from 70% in 2015 [2]. The variation in the user's demand concerning resolution, bit rates, frame rate or a combination of these makes the job of media professionals critical to managing it. It is not possible to store videos with all possible formats, resolutions, frame rates as it requires massive storage and computational resources. This process will also increase the financial cost of the video content providers. One of the solutions is to store some (e.g. popular) video in popular formats and transcode unpopular videos on demand [5]. The rapid growth of mobile devices, user preferences and networks have created a requirement for video transcoding into the appropriate specification such as resolution, quality, bit rate, video format, etc. and simultaneously ensuring different Quality of Services (QoS) such as delay. Converting a compressed video into another compressed video is termed as video transcoding [10]. But transcoding of videos in real-time is a time-consuming and challenging task as it holds a strict delay requirement.

Video transcoding not only reduces the video file size but also give an opportunity to select from an extensive set of options. It makes the video viewable across platforms, devices, and networks. Usually, transcoding requires a high-quality mezzanine file to start with and convert it into a form supported by the targeted device. Transcoding a large, high-definition video to a diverse set of screen sizes, bit rates, and quality requires a lot of time, computation, storage capacity. To overcome the difficulty associated with the transcoding process content providers, is using the cloud services. A user only needs to specify its requirements and subscribe the services provided by the cloud, and the rest of the task, i.e. time-consuming transcoding process will be performed using cloud resources at the back-end. The advantage of transcoding in the cloud is lower cost, virtually unlimited scalability, and elasticity to counter peak demand in real-time. The cloud transcoding solution allows video service providers to pay as they use, with the assurance of providing online support to handle unpredictable demands [1]. Cloud-based video transcoding reserve resources based on current workload to satisfy predefined QoS. However, online transcoding in the cloud has its challenges. The first key problem is the hard delay(streaming and transcoding) requirement. The second challenge is balancing resources available and the demand while ensuring cost and QoS constraints. The insufficient resource reservation for transcoding may cause delay of the video playback. Video transcoding in the cloud can be done in following ways: through a dedicated VM or using different VMs for different video segments simultaneously. The first approach requires a significant number of VMs for a large set of video stream, whereas the second method can transcode several video streams simultaneously reducing the number of VMs [5]. In this paper, we assume the second approach and discuss the implementation of a cloud-based platform for transcoding of videos.

The rest of the paper is organized as follows: Section 2 summarizes the related works. Cloud-based video transcoding architecture and resource allocation policy are presented in section 3 and 4, respectively. Section 5 analyzes the simulation results. The paper is concluded in section 6.

2 Literature Study

Researchers proposed and implemented various transcoding framework where video transcoding can be performed partially or on-demand. Few works on cloud-based transcoding is presented here. Researchers have worked on energy-efficient and real-time task allocation [8, 9]. Hui *et al.* address the cost issue of multi-version video on demand system in the cloud. The decision whether to store or transcode is made based on the popularity of video, storage, and computation cost [13]. Weiwen *et al.* proposed an energy-efficient algorithm to route transcoding jobs in the multimedia cloud [12]. Lei *et al.* presented an analytical model of a cloud-based online video transcoding system to predict the minimum resource reservation for specific QoS constraints (i.e. minimum system delay, targeted chunk size) [11]. Guanyu *et al.* used a partial transcoding scheme to minimize the operational (i.e. storage + computation) cost of content management in media cloud. Based on the user viewing pattern decision is made whether to cache or transcode online a video segment [4]. Xiangbo *et al.* introduced a cloud-based video streaming service (CVSS) architecture for on-demand transcoding of video streams using cloud resources. CVSS architecture gives a cost-efficient platform to streaming service providers for using cloud resources to meet QoS demands of video streams [6]. Zhenhua *et al.* designed a cloud transcoder to reduce the download time and improve the data transfer rate. Transcoding is executed based on video popularity and transcoder status, i.e. below the certain threshold [7]. Fareed *et al.* presented a prediction-based dynamic resource allocation algorithm to allocate and deallocate VMs for video transcoding service with the aim of achieving cost efficiency in Infrastructure as a Service cloud [5]. Kwei *et al.* implemented a cloud-based, scalable and cost-effective video streaming service platform to serve all the transcoding requests [3].

3 System Model

We propose the cloud-based on-demand video transcoding (CVT) framework as shown in Fig.1. The framework shows the sequence of actions taken place when a user request videos from a streaming service provider. The functionalities of various components are as follows: The *streaming service providers* like YouTube, Netflix accepts user's request and checks if required video is present in video repository or not. If the video is present in its desired format, then starts streaming the video. If the video is not in a format that is requested, online transcoding is done using cloud resources. To perform transcoding, video is divided into small chunks by the *video splitter* and then *transcode manager* map the video chunks to appropriate transcoding VMs based on certain QoS. The

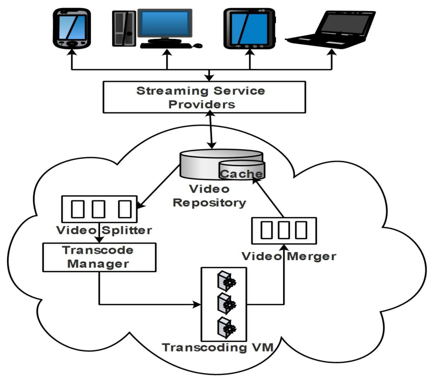

Fig. 1. Cloud-based Video Transcoder

queue formed by video streams near transcode manager has two parts: startup and batch. Start-up part of the queue consists of first few Group of Pictures (GOPs) of each video stream, and rest streams are in batch queue part. For a video request, video streams in startup queue are assigned to VM first and then the streams present in the batch queue. The *transcoding VM* transcode the source videos into targeted videos with desired video specification concerning format, resolution, quality, etc. with certain QoS constraints. Each transcoding VM is capable of processing one or more simultaneous transcoding task. *Video Merger* is used to place all the video streams in the right order to create the resulting transcoded stream. A copy of the transcoded video is stored in video repository to save time and computation cost. All the possible forms of popular and frequently accessed videos are stored in *cache storage*, a part of video repository. The unpopular video requested by the user is transcoded online and served to the user.

Algorithm 1: CVS Resource allocation algorithm

Input : $V_{DS} \leftarrow V_1, V_2, ..., V_n$: List containing n videos
$nI \leftarrow$ number of I frames of each video
$Q \leftarrow$ Queue contains 1^{st} video chunk, 2^{nd} video chunk, ... of $V_1, .., V_n$.
$size \leftarrow$ size of the video
$len_{sq} \leftarrow$ length of queue associated with startup queue(SQ)
$len_Q \leftarrow$ length of Q
$len_{bq} \leftarrow$ length of batch queue(BQ)
$maxTH \leftarrow$ Maximum threshold

Output : $nVMs \leftarrow$ *the number of VMs required*

Initialise: $nVMs \leftarrow 0$

$nIFrames \leftarrow nI\ of\ V_i$
$Enqueue\ len_{sq}\ no\ of\ video\ chunks\ from\ Q\ to\ SQ$
$Allocate\ SQ\ to\ VM$
$nVMs \leftarrow nVMs + 1$
$len_{bq} \leftarrow len_Q - len_{sq}$
for $i = 1\ to\ len_{bq}$ **do**
  $calculate\ avgQlen\ of\ VMs$ ▷ /*avgQlen() calculates the average of filled portion of queue of each VM*/
  if $avgQlen() \geq maxTH$ **then**
    $activate\ new\ VM$
    $nVMs = nVMs + 1$
  end
  $VM_{min} \leftarrow Select\ VM\ with\ shortest\ filled\ queue$
  $enqueue\ BQ[i]\ to\ VM_{min}$
end

4 Resource Allocation Policy

The proposed algorithm, Algorithm 1 predicts the number of VMs for transcoding the videos demanded by users. The videos are divided into smaller chunks at GOP level. Therefore the number of smaller chunks possible is equal to number of I frames present in a particular video. The queue Q is formed by taking i_{th} video chunk from each video and en-queuing them sequentially where $i = 1, 2, ...n$ and

n is the number of videos. In step 2 and 3, a start-up queue (SQ) is formed and allocated to VM. The start-up queue will transcode some initial chunks of videos to provide faster response to users. So we always have 1 or more VMs for transcoding. Once the start-up queue is completed, the corresponding VM can be used with other VMs for transcoding the batch queue (BQ). Before allocating a video chunk to any VM, the average load (queue length of each VM) of active VMs is calculated. If the average length is more than the threshold value, a new VM is activated. The VM with least load is selected (V_{min}) and the new video chunk is en-queued to V_{min}.

5 Simulation Results

We have uniformly selected videos from YouTube dataset collected from UCI repository- Online Video Characteristics and Transcoding Time Dataset Data Set for simulation. Out of 10 fundamental video characteristics, the attributes which we have used here are: video ID, duration, codec, number of I frames and total transcoding time. We have made certain assumptions to correlate to the result better.

1. Number of I frames in a particular video is same as the number of GOPs.
2. The transcoding time of each GOP by dividing total transcoding time with number of I frames.
3. The VMs allocated are considered as homogeneous.
4. The queue associated with each VM has same length (say 100). The maximum threshold value is assumed to be 70% of the VM's queue length.

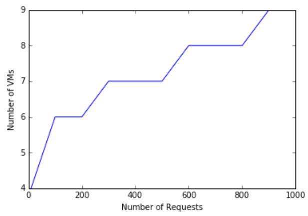

Fig. 2. Resource prediction by Proposed algorithm

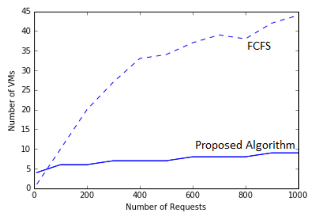

Fig. 3. comparison of proposed algorithm with FCFS

Fig. 2 demonstrates the number of VMs required for transcoding the corresponding number of requests. We see that the resource allocated for the users' requests are quiet cost effective as the number of VMs active are less because we perform the initial check whether to activate another VM as per need. In Fig. 3, there is a comparison of our proposed resource allocation algorithm with existing scheduling algorithm. It is observed that the number of VMs required in FCFS is quite high in comparison to the proposed algorithm.

6 Conclusion

In this paper, we propose an on-demand CVT architecture that provides a cost-efficient platform to transcode video streams. The dynamic resource allocation scheme present in CVT predicts the appropriate number of VMs so that resource provisioning cost is reduced. For cost efficiency, a video is broken into several video streams so that multiple streams can be transcoded on a single VM and thus reducing the VM requirement. This architecture can be useful for video streaming providers to utilize cloud resources and improve user's satisfaction with low cost.

References

1. http://download.sorensonmedia.com/pdfdownloads/lowres/whitepaper.pdf. 2011.
2. Forecast and methodology, 2015-2020 white paper. *Cisco Visual Networking Index*, 2016.
3. Kwei-Bor Chen and Hong-Yi Chang. Complexity of cloud-based transcoding platform for scalable and effective video streaming services. *Multimedia Tools and Applications*, pages 1–18, 2016.
4. G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu. Towards cost-efficient video transcoding in media cloud: Insights learned from user viewing patterns. *IEEE Transactions on Multimedia*, pages 1286–1296, 2015.
5. F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius. Prediction-based dynamic resource allocation for video transcoding in cloud computing. In *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 254–261. IEEE, 2013.
6. X. Li, M. A. Salehi, M. Bayoumi, and R. Buyya. Cvss: A cost-efficient and qos-aware video streaming using cloud services. In *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 106–115. IEEE, 2016.
7. Zhenhua Li, Yan Huang, Gang Liu, Fuchen Wang, Zhi-Li Zhang, and Yafei Dai. Cloud transcoder: Bridging the format and resolution gap between internet videos and mobile devices. In *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 33–38. ACM, 2012.
8. S. K. Mishra, R. Deswal, S. Sahoo, and B. Sahoo. Improving energy consumption in cloud. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2015.
9. S. Sahoo, S. Nawaz, S. K. Mishra, and B. Sahoo. Execution of real time task on cloud environment. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–5, 2015.
10. Sampa Sahoo, Bibhudatta Sahoo, and Ashok Kumar Turuk. An analysis of video transcoding in multi-core cloud environment. 2017.
11. L. Wei, J. Cai, C. H. Foh, and B. He. Qos-aware resource allocation for video transcoding in clouds. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 49–61, 2017.
12. W. Zhang, Y. Wen, J. Cai, and D. O. Wu. Toward transcoding as a service in a multimedia cloud: Energy-efficient job-dispatching algorithm. *IEEE Transactions on Vehicular Technology*, pages 2002–2012, 2014.
13. H. Zhao, Q. Zheng, W. Zhang, B. Du, and H. Li. A segment-based storage and transcoding trade-off strategy for multi-version vod systems in the cloud. *IEEE Transactions on Multimedia*, pages 149–159, 2017.