

# Cost Efficient Repository Management for Cloud-Based On-Demand Video Streaming

Mahmoud Darwich<sup>\*†</sup>, Ege Beyazit<sup>\*</sup>, Mohsen Amini Salehi<sup>†‡</sup>, Magdy Bayoumi<sup>\*</sup>

<sup>\*</sup>Center for Advanced Computer Studies (CACCS)

<sup>‡</sup>High Performance Cloud Computing (HPCC) Laboratory

<sup>†</sup>School of Computing and Informatics

University of Louisiana at Lafayette, Louisiana 70503

Email: {mkd1007, exb6143, amini, mab}@louisiana.edu

**Abstract**—Video transcoding is the process of converting a video to the format supported by the viewer’s device. Video transcoding requires a huge storage and computational resources, thus, many video stream providers choose to carry it out on the cloud. Video streaming providers generally need to prepare several formats of the same video (termed pre-transcoding) and stream the appropriate format to the viewer. However, pre-transcoding requires enormous storage space and imposes a significant cost to the stream provider. More importantly, pre-transcoding proven to be inefficient due to long-tail access pattern to video streams in a repository. To reduce the incurred cost, in this research, we propose a method to partially pre-transcode video streams and re-transcode the rest of it in an on-demand manner. We will develop a method to strike a trade-off between pre-transcoding and on-demand transcoding of video streams to reduce the overall cost. Experimental results show the efficiency of our approach, particularly, when a high percentage of videos are accessed frequently. In such repositories, the proposed approach reduces the incurred cost by up to 70%.

**Index Terms**—cloud, storage, video stream, partial pre-transcoding, re-transcoding

## I. INTRODUCTION

Video streaming through the Internet has become a common practice. Viewers stream videos on variety of devices, from large screen TVs and desktops to tablets and smart-phones. Based on the Global Internet Phenomena Report [1], video streaming currently constitutes around 64% of the U.S. Internet traffic and it is rocketing to 80% of the whole Internet traffic by 2019 [2].

Video contents, either in the form of Video On Demand (VOD) (e.g., YouTube<sup>1</sup> or Netflix<sup>2</sup>) or live-streaming (e.g., Livestream<sup>3</sup>), need to be converted (i.e., transcoded) based on the device characteristics of viewers. That is, the original video has to be converted to a supported characteristics (e.g., frame rate, resolution, and network bandwidth) of the viewers’ devices [3].

Video transcoding is a compute-intensive and time-consuming process. To overcome the computational demand of transcoding, Video Stream Providers (VSP) extensively utilize cloud computing services [4]. To make the VOD streams

readily available for viewers, VSPs commonly carry out the transcoding operation of VODs in an off-line manner. That is, they transcode and store multiple formats of the same video in the cloud to satisfy the requirements of viewers with heterogeneous display devices. We call this approach *pre-transcoding*. In practice, VSPs such as Netflix have to pre-transcode 40 to 50 different formats of a single video [5] and store them in their repositories. This imposes a significant cost overhead to VSPs [6], [8].

Recent studies show that accessing videos of a VSP follows a long tail distribution [7]. That is, there are few videos that are accessed very frequently while there is a huge portion of videos that are rarely accessed. Thus, research works have been undertaken (e.g., [4], [8]) to alleviate the cost overhead of pre-transcoding by transcoding rarely-accessed videos in an on-demand (i.e., lazy) manner. In this manner, one or few formats of a video is stored and transcoding is performed on-the-spot upon request to access a format of video that is not already pre-transcoded. This has become feasible with the enormous computational capacity clouds offer. We term the lazy transcoding of videos as *re-transcoding* and storing of videos as *pre-transcoding*.

Re-transcoding induces the computational cost of VSPs which is generally more expensive than the storage cost [9], this is because computation power are charged in on hourly basis in cloud. Therefore, the re-transcoding approach would be beneficial to VSPs, only if it is applied on the rarely accessed videos. Conversely, if it is applied on *frequently accessed videos* (FAVs), it increases the cost overhead significantly as we have to pay for every time the video stream is transcoded.

The challenge is how to achieve partial pre-transcoding? That is, which parts of the video should be re-transcoded and which part pre-transcoded?

To address these challenges, in this paper, we propose a method to perform pre-transcoding on a portion of the video.

In summary, the contributions of this paper are: **(A)** Proposing method to reduce the incurred cost of using cloud services through pre-transcoding, re-transcoding, or partially-pretranscoding videos in the repository. **(B)** Analyzing the impact of our proposed method when the rate of access to video streams in a repository varies.

Experimental results demonstrate that our proposed partial-

<sup>1</sup><https://www.youtube.com>

<sup>2</sup><https://www.netflix.com>

<sup>3</sup><https://livestreams.com>

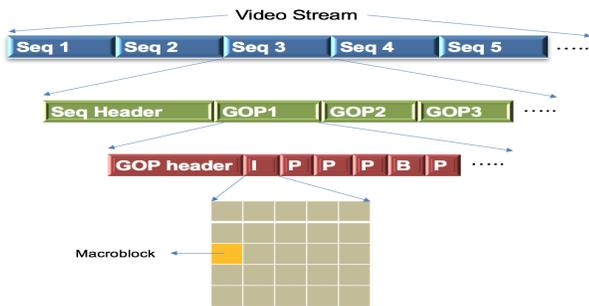


Fig. 1: Video Stream Structure

transcoding method can reduce the cost overhead of VSPs significantly. The research outcome of this paper can help VSPs to reduce their cost overhead without losing Quality of Service (QoS) demanded by their viewers.

The rest of the paper is organized as follows: section II provides some background on video streaming and transcoding. Section III presents the partial pre-transcoding method. Experiment setup is detailed in section IV. Experimental results are discussed in section V, related works are presented in section VI and finally section VII concludes the paper.

## II. BACKGROUND

### A. Video Stream Structure

A video stream is a set of sequences as shown in Fig. 1. Each sequence is built of several *Group Of Pictures (GOPs)*. The first block of a sequence is called a sequence header that contains some meta-data about that sequence. Also, each GOP is constructed of a GOP header followed by several frame types, starting with I (intra) frame, followed by a P (predicted) and B (bi-directional) frames. Each frame is further comprised of slices that are formed from macroblocks (MB) [17].

As each GOP can be processed independently, video transcoding operation is commonly achieved at the GOP level [17]. That is, each GOP is considered as a unit for processing or pre-transcoding.

### B. Cloud Services for Video Stream Transcoding

Cloud providers offer different services in an on-demand manner and charge their users in a pay-as-you-go manner. A cloud-based video streaming system utilizes different cloud services as follows:

- **Computational Services:** Computational services in clouds are generally provided through Virtual Machines (VM) and users are charged in an hourly basis.
- **Storage Services:** Users are charged for storage services based on the volume of their data stored on the cloud usually on a monthly basis.
- **Content Delivery Network (CDN) services:** CDN is a technology that reduces the delay to access different static content types, including video streams, through the Internet. CDN technology replicates the content (*e.g.*, video content) in different geographical areas to minimize the network travel time of content to users [10], [11].

Amazon Web Services<sup>4</sup> is a major cloud provider and offers all the foregoing cloud services with a high reliability. Although this study is independent of AWS technology and can work on any of cloud provider, we consider AWS services, charging model, and costs for our evaluations.

Amazon Elastic Compute Cloud (Amazon EC2) provides computational services in form of VMs. It offers various types of VMs to cover different computational demands. General purpose  $t2$ -small VM type is the most common service used for different type of processing and we utilize this VM type for our evaluations as well. The hourly cost of  $t2$ -small VM is \$0.026. Amazon Simple Storage Service (Amazon S3) is the storage service of Amazon cloud. Amazon S3 costs \$0.03 for each Gigabyte of stored data in a month. Amazon also offers the CDN service (called CloudFront<sup>5</sup>). It delivers the content to users (*i.e.*, viewers) through a worldwide network of data centers with minimum delay [12]. Amazon charges \$0.085 for each Gigabyte of data uses CloudFront for the first 10TB per month.

It is worth noting that CDN services are required, in addition to storage services to perform pre-transcoded cloud-based video streaming. However, the CDN cost is not applied when we provide on-demand video transcoding service.

## III. PROPOSED PARTIAL PRE-TRANSCODING METHOD

### A. Video Streams Access Pattern

As re-transcoding cost is incurred every time a video stream is transcoded, it is crucial to know the access pattern to videos. Previous studies show that accessing video streams follow a long tail distribution. Other observations (*e.g.*, [15]) reveal that, within each video stream, the beginning segments (GOPs) are watched more frequently than the rest of the video streams.

Miranda *et al.* [15] show that the distribution of the views within a video stream follows a long tail distribution. More specifically, they show that the distribution of accesses to GOPs of a video stream can be expressed by Power-law [18] model. Therefore, if a video stream is accessed  $V$  times, then we can estimate the number of times each GOP is viewed. Let  $GOP_i$  as the  $i$ th GOP in a video stream, then the number of views for  $GOP_i$ , denoted  $P_i$ , is estimated based on Equation 1. In this equation,  $\alpha$  is a constant with value 0.1.

$$P_i = \frac{V}{GOP_i^\alpha} \quad (1)$$

### B. Storage Cost of Video Streams

The storage cost of a video stream depends on the video size, and the cloud storage unit price. The equation of cloud storage cost for video  $v$  is defined as:

$$C_S = \frac{S_v \cdot P_S}{2^{10}} \quad (2)$$

where  $S_v$  is the size of video  $v$  in MB,  $P_S$  is the storage unit price in dollar per GB, the term  $2^{10}$  is used to convert the

<sup>4</sup><https://aws.amazon.com/>

<sup>5</sup><https://aws.amazon.com/cloudfront/>

video size unit from MB to GB. Equation 2, however, might be extended to compute the storage cost of each GOP in the video stream. The equation of storage cost for  $GOP_i$  is  $C_{S_i} = \frac{P_S \cdot S_{GOP_i}}{2^{10}}$ , where  $S_{GOP_i}$  is the size of  $GOP_i$ .

### C. Transcoding Cost of Video Streams

The transcoding cost is the cost of using virtual machines (VMs) that depends on the time span of utilizing VMs and the type of VM utilized for transcoding. Let  $P_T$  the cost of using VM for an hour, and  $\tau_v$  the estimated transcoding time of video  $v$  in seconds. Then, the cost of transcoding of video  $v$ , denoted  $C_T$ , is obtained using Equation 3.

$$C_T = \frac{P_T \cdot \tau_v}{3600} \quad (3)$$

It is noteworthy that Equation 3 determines the cost for one time re-transcoding of a video. However, if a video is re-transcoded  $V$  times, the total cost would be  $V \cdot C_T$ .

Estimation of transcoding execution time ( $\tau_v$  in Equation 3) can be obtained based on historic transcoding execution times of the video in the past<sup>6</sup>. In particular, the estimated transcoding time of video  $v$  is the sum of transcoding time of all GOPs in that video, *i.e.*,  $\tau_v = \sum_{i=1}^m \tau_i$ , where  $\tau_i$  is the estimated transcoding time of  $GOP_i$  and  $m$  is the total number of GOPs in video  $v$ .

Accordingly, Equation 3 can be extended to compute the transcoding cost of each GOP in the video stream. The equation of transcoding cost for  $GOP_i$  is represented by  $C_{T_i} = \frac{P_T \cdot \tau_i}{3600}$ , where  $\tau_i$  is the estimated transcoding time of  $GOP_i$ .

### D. Algorithm 1

The goal of this algorithm is to minimize the incurred cost of using cloud services for VOD transcoding. For that goal, the algorithm is executed for each video in the VSP repository periodically (*e.g.*, monthly). The pseudo code for this algorithm is presented in Algorithm 1. It receives the video size, estimated video transcoding time, storage unit price, transcoding unit price, and number of views of the video in the last time period (*e.g.*, last month) as input values. The output of the algorithm is whether to pre-transcode the whole video or re-transcode it upon request.

The algorithm calculates the cost of storage and transcoding according to what was discussed in Equation 2 and 3 ( see steps 1 and 2 Algorithm 1). We calculate the ratio of video storage cost to video transcoding cost in step 6 (called *cost ratio*). When the ratio is less than or equal 1, it means that the cost of pre-transcoding is less than or equal the re-transcoding cost, thus, we store that video format, otherwise, we call Algorithm 2 to decide about partial pre-transcoding of the video. It is worth noting that for a new video  $V = 0$ , thus, we postpone its transcoding to when it is requested.

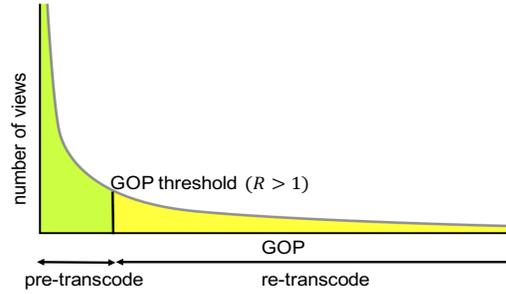


Fig. 2: Long tail distribution of GOPs views in video stream and partial pre-transcoding a video stream based on GOP threshold

### E. Algorithm 2

When the cost ratio for a video is greater than 1, Algorithm 2 is called to possibly pre-transcode a portion of the video (termed partially pre-transcoded video). It is proven that the access pattern of GOPs within a video stream follows a long tail distribution [15]. That is, the first GOPs of a video stream are watched more often than the rest. Thus, the cost ratio of GOPs constantly increases for the later GOPs in the video stream. Accordingly, it is possible to find a boundary GOP so that all GOPs before it need to be pre-transcoded and all GOPs after that re-transcoded. We call this dividing point as GOP threshold ( $GOP_{th}$ ). Formally,  $GOP_{th}$  is defined as the first GOP of a video stream that has its cost ratio greater than one.

Algorithm 2 receives GOPs of a video stream, GOP size, estimated transcoding time of GOP, storage unit price, transcoding unit price and number of requests for the video stream in the last time period as input values.

Algorithm 2 searches for the GOP threshold by calculating the cost ratio of each GOP sequentially, starting from the first GOP in the video stream (step 1 to 5 in Algorithm 2). After finding the GOP threshold, Algorithm 2 pre-transcodes the GOPs before  $GOP_{th}$  and re-transcodes the GOPs after it as shown in Algorithm 2

Fig. 2 illustrates how Algorithm 2 functions based on the long tail access pattern to GOPs within a video stream. Once we find the GOP threshold, all the GOPs with cost ratio less than 1 (*i.e.*,  $R_i < 1$ ) are pre-transcoded and the rest of GOP are re-transcoded.

## IV. EXPERIMENT SETUP

### A. Videos Synthesis

VSPs have large video streams repository. However, we do not have access to such repositories. Downloading a large quantity of videos and then transcoding them is a long and costly process. Therefore, we simulate large repositories by synthesizing videos.

To accurately synthesize videos, we need to know the distribution of characteristics of videos. Specifically, GOP size,

<sup>6</sup>As we are dealing with VOD, we expect that videos have been watched, thus, transcoded before. However, this is not the case when we deal with live stream videos. In that case such historic execution information is not available.

---

**Algorithm 1:** Fully pre-transcode or fully re-transcode a video

---

**Input :**

Size of video  $v$ :  $S_v$   
 Transcoding time of video  $v$ :  $\tau_v$   
 Transcoding unit price:  $P_T$   
 Storage unit price:  $P_S$   
 Number of views in the last time period:  $V$

**Output:** fully pre-transcoding or fully re-transcode a video

- 1 Calculate cost of transcoding of  $v$ :  $C_T \leftarrow \frac{P_T \cdot \tau_v}{3600}$
  - 2 Calculate cost of storage of  $v$ :  $C_S \leftarrow \frac{S_v \cdot P_S}{2^{10}}$
  - 3 **if**  $V = 0$  **then**
  - 4     Re-transcode the whole video  $v$
  - 5 **end**
  - 6 **if**  $\frac{C_S}{V \cdot C_T} \leq 1$  **then**
  - 7     pre-transcode video  $v$
  - 8 **else**
  - 9     Perform partial pre-transcoding using Algorithm 2
  - 10 **end**
- 

---

**Algorithm 2:** Partial pre-transcoding a video

---

**Input :**

video  $v$  with  $m$  GOP  
 Size of  $GOP_i$ :  $S_i$   
 Transcoding time of video  $GOP_i$ :  $\tau_i$   
 Transcoding unit price:  $P_T$   
 Storage unit price:  $P_S$   
 Views number in the last time period :  $V$

**Output:** Partial pre-transcoding a video

- 1 For each  $GOP_i$  in video stream  $v$
  - 2     Estimated number of views for  $GOP_i$ :  
 $P_i \leftarrow V \cdot GOP_i^{-\alpha}$
  - 3     Calculate cost of transcoding of  $GOP_i$ :  
 $C_{T_i} \leftarrow \frac{P_T \cdot \tau_i}{3600}$
  - 4     Calculate cost of storage of  $GOP_i$ :  $C_{S_i} \leftarrow \frac{S_i \cdot P_S}{2^{10}}$
  - 5     Calculate cost ratio of  $GOP_i$ :  $R_i \leftarrow \frac{C_{S_i}}{P_i \cdot C_{T_i}}$
  - 6     **if**  $R_i > 1$  **then**
  - 7          $GOP_{th} \leftarrow GOP_i$
  - 8         Break;
  - 9     **else**
  - 10        Continue;
  - 11     **end**
  - 12 Pre-transcode  $GOP_1$  to  $GOP_{th-1}$
  - 13 Re-transcode  $GOP_{th}$  to  $GOP_m$
- 

	GOP size (Kbytes)	number of GOPs
Mean	655.08	1262.79
Standard Deviation	201.44	271.46
Standard error mean	0.57	26.74
Upper 95% Mean	656.20	1315.85
Lower 95% Mean	653.96	1209.74
Total no. of GOP	124593	130068
Min. GOP size	1.91	580
Max. GOP size	2192.65	2018

TABLE I: statistics of GOP size and number of GOPs in a video repository

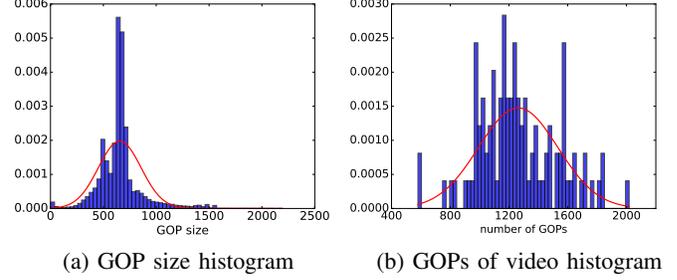


Fig. 3: distribution of GOP size and numbers of GOPs in videos

GOP transcoding time, and number of GOPs for each video are required. We synthesize videos based on our repository that includes 103 videos of a wide range of categories including movies, animation, sports, documentary, news, and music. the size of the videos are between 400MB and 2.5GB. In order to extract the characteristics of those videos, we transcode them by implementing the method in [4] on a machine with 2.8 GHz Intel Core i5, 8GB RAM, and 1TB Hard Disk. We analyze the characteristic data extracted from videos and observe that GOP size and number of GOPs follow a Gaussian curve, shown in Fig. 3a and 3b. The parameters of the Gaussian distributions are given in table I. Additionally, Li *et al.* [19] proved through a regression analysis that there is a linear relationship between GOP size and its transcoding time. By implementing the same method we derive a linear equation between GOP sizes and their corresponding transcoding times for all downloaded videos and the plot of the analysis is shown in Fig. 4.

Having the statistics of GOP size, number of GOPs, and the linear equation for GOP transcoding times, we generate 50,000 videos with 112.98 TB size for our synthesized repository.

### B. View Pattern for Synthesized Videos

According to [7], we model the access pattern to repository by applying Weibull distribution to generate long tail access pattern. The frequently accessed videos (FAVs) that have high access rates, are located at the beginning of the Weibull curve and those have less views are in the long tail. We determine the percentage of FAVs number in the repository based on the shape and scale coefficients  $\alpha$  and  $\beta$  respectively of the Weibull distribution. As alpha decreases, the number of FAVs increases in the repository and vice versa. The interval range of  $\alpha$  used in the experiment is  $[0.4 : 2.4]$  while  $\beta = 1$

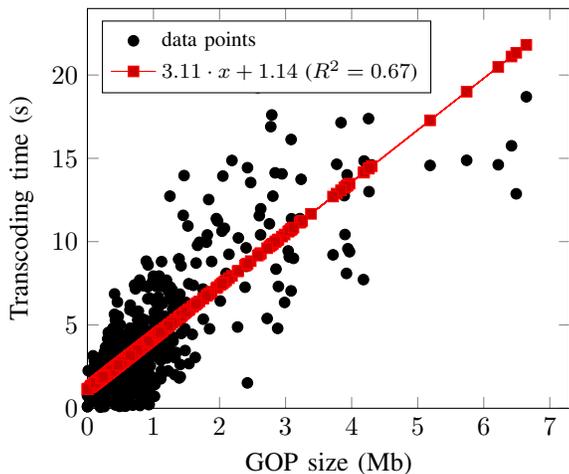


Fig. 4: Linear regression analysis of GOP size and transcoding time

### C. Storage and Computation Prices

The prices of storage and computation used in the experiments are modeled based on Amazon Web Service (AWS) prices as shown in Table II

Transcoding	storage	CDN
t2-small	s3	CloudFront
\$0.026 /hour	\$0.03 GB/month	\$0.085 GB/month

TABLE II: AWS computation, storage, and Content Delivery Network (CDN) services and their prices in US dollar \$

### D. Baseline Methods for Comparison

To evaluate our proposed algorithms, we compare them against two baseline methods that are *Fully pre-transcoding* (that stores the whole video streams) and *Fully re-transcoding* (that re-transcodes all video streams upon request). These two methods ignore the metadata of processing and storage costs of the video streams.

## V. EXPERIMENTAL RESULTS

### A. Impact of FAVs percentage in repository

We create several synthesized repositories of video streams to illustrate the performance of the proposed algorithms. Each synthesized repository is made of total 50,000 videos with a percentage of FAVs. The shape parameter in of Weibull distribution ( $\alpha$ ) controls the percentage of FAVs in the repositories. The percentage of FAVs based on  $\alpha$  and  $\beta$  are shown in Table III. The average view of videos in the repositories is 1.99. We conduct each experiment 10 times and report the mean and 95% confidence interval.

Fig. 5a shows the total cost of fully pre-transcoding, fully re-transcoding and partial pre-transcoding methods when the percentage of FAVs vary. When  $\alpha$  is small, the percentage of FAVs increases and they, in turn, increase the cost of transcoding process. As illustrated in the figure, the fully storage cost does not change and it stays constant even when

	$\alpha$	FAVs
$\beta = 1$	0.4	30%
	0.6	25%
	1	20%
	1.4	15%
	1.8	10%
	2.4	5%

TABLE III: Percentage of FAVs in the repository varies by changing  $\alpha$  in the Weibull distribution

changing the percentage of FAVs, because the fully storage cost is independent of the number of views of video streams.

The experiment shows that our proposed method always reduces the incurred cost comparing to fully re-transcoding method by up to 70% when 30% of the repository is FAVs. Also, comparing to the fully pre-transcoding method, our proposed approach reduces the cost up to 60%. As the percentage of FAVs increases, our proposed method can reduce the total incurred cost significantly.

Similarly, in Fig. 5b, we includes the cost of Content Delivery Network (CDN) to the cloud storage cost. The proposed partial re-transcoding method reduces the cost up by 66%. In fact, when we consider CDN, the total cloud storage cost is increased. We note that as the cost ratio is increased, the number FAVs which are dependent on this ratio, decreases in the repository.

### B. Impact of average of number of views

In this experiment, we increase the average number of views of FAVs without increasing percentage of FAVs, as shown in Fig. 5c. In this experiment, the repository includes 20% of FAVs. The average number of views of FAVs is multiplied by constant 2, 3, 4, and 5 and the remaining videos in the repository are divided by the same constants.

The cost of fully re-transcoding increases, because FAVs have high views which has direct impact on the re-transcoding cost. The proposed method outperforms the fully pre-transcoding and fully re-transcoding methods. As the views of videos streams increases, the total cost of partial pre-transcoding method reduces.

## VI. RELATED WORK

Gao *et al.* [16] proposed a scheme that partially transcodes video contents in the Cloud. Their approach aims to store the first segments of a video contents which are more frequently viewed, while transcode the remaining video contents online when requested, resulting storage and transcoding computation cost efficiency. Their system model is based on a partial transcoding of a video segments and a dynamic storing system that is adapted to the video access rate change. They designed online algorithm to optimize the performance of their scheme. They stated that the viewers play 20% of the video duration. they adopted the video access pattern by implementing the truncated exponential distribution to represent the video view rate. They demonstrated that their method reduces 30% of operational cost by compared to storing the whole video.

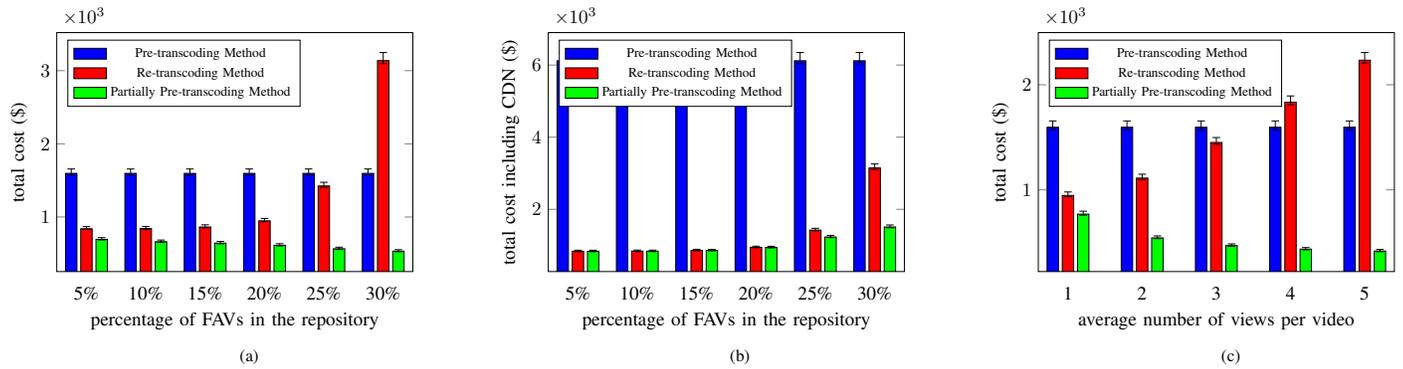


Fig. 5: Incurred costs of fully pre-transcoding, fully re-transcoding, and partially pre-transcoding methods: (5a) when the percentage of FAVs varies in the repository; (5b) when the CDN cost is added to the cloud storage cost; And (5c) when the average view for videos in the synthesized repository is increased.

Zhao *et al.* [14] proposed an approach to trade-off between computation and storage costs and to minimize the cost for multi version videos. They utilized the transcoding weight graph which is the transcoding relationships among versions of a video, along with the popularity of those different versions of the video. Based on the popularity and transcoding relationships among different video versions, their method decides which versions of a video should be stored in the repository or re-transcoded on demand. Their results shows reduction in the cost when compared to storing all versions.

Jokhio *et al.* [13] developed a strategy to strike a trade off between the computation and storage costs of a video. They estimated the computation cost, the storage cost, and the video popularity information of individual transcoded videos and then utilized this information to make decisions on how long a video should be stored or how frequently it should be re-transcoded from a given source video. They compared their proposal to semisynthetic and realistic load patterns. Their results indicated that their strategy is more cost-efficient than the two intuitive strategies.

## VII. CONCLUSION

In this paper, we propose a method to reduce the incurred cost of using cloud services through pre-transcoding, re-transcoding, or partially pre-transcoding of videos in the videos repository. We also analyze the impact of our proposed method when the rate of access to video streams in a repository varies. Experiment results show that as the percentage of FAVs in the repository increases, the proposed method reduces the total incurred cost significantly. In particular, by increasing the number of views of FAVs, our method reduces the total incurred cost by up to 70%. Our future work will focus on the prediction of the number of views of a video stream by applying machine learning techniques. The method will operate based on views statistics over the past and thus we can develop offline algorithms to estimate the cost of video stream transcoding in the cloud. This method will help VSPs to optimize the cost of using cloud resources.

## REFERENCES

- [1] G. I. P. Report, <https://www.sandvine.com/trends/global-internetphenomena/>, accessed Oct. 1, 2016.
- [2] C. V. N. Index, Forecast and methodology, 2014-2019, 2015
- [3] Ahmad, Ishfaq, et al. "Video transcoding: an overview of various techniques and research issues." *IEEE Transactions on multimedia 7.5* (2005)
- [4] Li, Xiangbo, Mohsen Amini Salehi, and Magdy Bayoumi. "Cloud-based video streaming for energy-and compute-limited thin clients." *the Stream2015 Workshop at Indiana University*, 2015.
- [5] <http://techblog.netflix.com/2012/12/videos-of-netflix-talks-at-aws-reinvent.html>
- [6] Li, Xiangbo, Mohsen Amini Salehi, and Magdy Bayoumi. "VLSC: Video Live Streaming Using Cloud Services." *6th IEEE International Conferences on Big Data and Cloud Computing (BDCloud)*, 2016.
- [7] N. Sharma, D. K. Krishappa, D. Irwin, M. Zink, and P. Shenoy "GreenCache: augmenting off-the-grid cellular towers with multimedia caches." *Proceedings of the 4th ACM Multimedia Systems Conference*, 2013.
- [8] X. Li, M. A. Salehi, M. Bayoumi, and R. Buyya, CVSS: A Cost- Efficient and QoS-Aware Video Streaming Using Cloud Services, in *Proceedings of the 16th ACM/IEEE International Conference on Cluster Cloud and Grid Computing*, ser. CCGrid 16, May 2016.
- [9] <https://aws.amazon.com/ec2/pricing/on-demand/>
- [10] Saroiu, Stefan, et al. "An analysis of internet content delivery systems." *ACM SIGOPS Operating Systems Review* 36 (2002)
- [11] Vakali, Athena, and George Pallis. "Content delivery networks: Status and trends." *IEEE Internet Computing* 7.6 (2003)
- [12] <http://docs.aws.amazon.com/>
- [13] F. Jokhio, A. Ashraf, S. Lafond and J. Lilius, "A Computation and Storage Trade-off Strategy for Cost-Efficient Video Transcoding in the Cloud," *39th Euromicro Conference on Software Engineering and Advanced Applications*, 2013
- [14] Zhao, Hui, *et al.* "A version-aware computation and storage trade-off strategy for multi-version VOD systems in the cloud." *20th IEEE Symposium on Computers and Communication (ISCC)*, 2015.
- [15] Miranda, Lucas CO, Rodrygo LT Santos, and Alberto HF Laender. "Characterizing video access patterns in mainstream media portals." *Proceedings of the 22nd International Conference on World Wide Web*, 2013.
- [16] 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," in *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1286-1296, Aug. 2015.
- [17] Jokhio, Fareed, *et al.* "Analysis of video segmentation for spatial resolution reduction video transcoding." *International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, 2011.
- [18] Newman, Mark EJ. "Power laws, Pareto distributions and Zipf's law." *Contemporary physics* 46.5 (2005)
- [19] X. Li, Yamini Joshi, M. K. Darwich, B. Landreneau, M. A. Salehi, and M. Bayoumi "Performance Analysis and Modeling of Video Transcoding Using Heterogeneous Cloud Services" *University of Louisiana at Lafayette, LA, Technical Report*, 2016