COST OPTIMAL VIDEO TRANSCODING IN MEDIA CLOUD: INSIGHTS FROM USER VIEWING PATTERN

Guanyu Gao[1]*, Weiwen Zhang*[1]*, Yonggang Wen*[1]*, Zhi Wang*[2]*, Wenwu Zhu*[3]*, Yap Peng Tan*[1]

[1]Nanyang Technological University, Singapore
[2]Graduate School at Shenzhen, Tsinghua University, China
[3]Tsinghua University, China
{ggao001, wzhang9, ygwen, eyptan}@ntu.edu.sg, {wangzhi@sz., wwzhu@}tsinghua.edu.cn

ABSTRACT

Video transcoding has been touted as an enabling technology to support growing media consumption over heterogenous devices. However, on-line transcoding could incur tremendous, if not prohibitive, cost in deploying or renting resources. In this research, we leverage an insight into the viewing pattern of video consumers to reduce the operating cost of video transcoding services. Specifically, it has been reported that viewers tend to terminate their session before the whole video is watched. As such, it is not cost-efficient for service providers to store or transcode all segments of the videos. Built upon this insight, we propose a partial transcoding scheme for content management in a media cloud to reduce the operating cost. Particularly, each content is split into multiple segments and stored in different files of varying playback rates. Some of the segments are stored in cache, resulting in storage cost; while some are transcoded in real-time in case of cache miss, resulting in computing cost. We aim to minimize the long-term operational cost by determining the number of segments for each playback rate to be cached or transcoded in real-time. We formulate this partial transcoding scheme as a constrained integer optimization problem. Leveraging Lagrangian relaxation and a subgradient method, we obtain the approximate solution to the integer program. Numerical results indicate that our proposed partial transcoding scheme can save more than 30% of operational cost, compared with a brute-force scheme of caching all the segments.

***Index Terms*—** Partial transcoding, user viewing pattern, media cloud, viewer behavior

1. INTRODUCTION

Mobile video, owning to the exploding popularity of mobile devices, has become one of the dominant contributors to mobile data traffic. According to Cisco's prediction, global mobile video will increase 16-fold between 2012 and 2017, which accounts for over 66% of total mobile data traffic [1]. However, limited bandwidth and unstable wireless channel inherently deteriorate the user experience, triggering a tussle between the growing demand of the mobile video traffic and the quality of network service.

Video transcoding has been touted as an enabling technology to support growing media consumption over heterogenous devices. [2] aimed to maximize the QoE for mobile users, by storing content copies transcoded in different bit rate. [3] proposed a distributed transcoding platform to reduce transcoding time for users. Those two works only focus on improving user experience of client side, without considering the operational cost of the content provider. Nevertheless, it was reported by [4] that it would cost millions of pounds for content providers to transcode and store a large number of content, including those seldom watched videos. Moreover, it is observed that only 10% of the most popular videos account for almost 80% of total views [5]; and for 60% of the videos, only less than 20% of their duration is viewed, most of users abort viewing within 40 seconds [6, 7, 8]. This user viewing pattern reveals that users consume only a small fraction of each video, which should attract content providers' attention when designing the content management system.

Based on the insight from user viewing pattern, we propose a partial transcoding scheme for content management in a media cloud. Any original video file is split into a set of segments, each of which has a fixed playback duration. When the user consumes some contents, it sends a request to the streaming engine with a required streaming rate. If the requested segment in the playback rate is available, users can consume the content immediately, resulting in storage cost; otherwise, a real-time transcoding is conducted, resulting in transcoding cost. The more segments are cached, the less transcoding tasks will be conducted, and vice versa. Therefore, there exists a tradeoff in optimizing the total cost.

Using the partial transcoding scheme, we aim to minimize the long term operational cost, including storage cost and computing cost, while satisfying the storage constraint and computation constraint of the streaming engine. We formulate the partial transcoding scheme as a constrained optimization problem, which is an integer program. Leveraging Lagrangian relaxation, we transform the original optimization

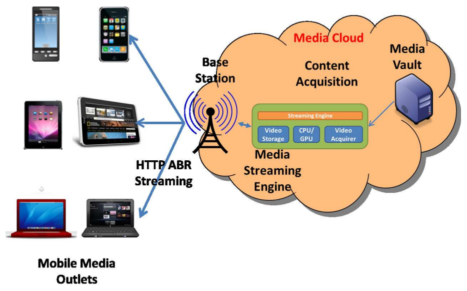


Fig. 1. A schematic diagram for streaming system: contents are transcoded into a set of files with different playback rates, and cached in the streaming engine.

problem and obtain the approximate solution. Particularly, we adopt subgradient method to achieve the optimal Lagrange multiplier. Numerical results show that i) the approximate solution is close to the optimal solution; ii) our proposed scheme can save more than 30% of operational cost, compared with all-segments-stored scheme; iii) our proposed scheme is robust to the change of the request arrival rate and more adaptive than the full transcoding scheme without segmentation.

To the best of our knowledge, this is the first work to adopt the partial transcoding scheme under the user viewing pattern to save operational cost from the economic perspective. Leveraging user viewing pattern, our proposed partial transcoding scheme can reduce the storage cost and computing cost for cloud content providers.

The rest of the paper is organized as follows: Section 2 presents system model and problem formulation. Section 3 provides an approximation algorithm for solving the problem. The experiment results are presented in Section 4. Finally, Section 5 concludes the paper and discusses the future work.

2. SYSTEM MODEL AND PROBLEM FORMULATION

2.1. System Architecture

This subsection presents our proposed system architecture for media cloud with video transcoding capability, serving a wide range of media outlets. This reference architecture is based on real experience with one of the leading players in China.

Our proposed cloud-based streaming system is illustrated in Fig. 1, the system consists of three parts, including media vault, media streaming engine and mobile media outlets. The functionalities of each modules are explained as follows:

Media Vault: it stores media files in their original formats and all the transcoded copies of alternative playback rates.

Media Outlet: it provides a display function for the viewers to consume the content and negotiates with the streaming engine for a proper playback rate.

Media Streaming Engine: it renders the content to media outlets in the corresponding playback rate, which is determined by the physical capability of the media outlet and the network status. The engine obtains content files from three alternative resources. First, in its local storage, a partial set of content files in various playback rates are stored for consumption. Second, a full copy of its original content is stored in local storage and can be transcoded in a real-time manner, by the CPU/GPU in the streaming engine, into the requested format. Finally, in rare case, it can retrieve content files of transcoded formats from the media vault. In this paper, we only focus on the first two cases. Specifically, an original video file is split into a set of sub-files, each of which consists of video content of a fixed playback duration (e.g., 2 seconds). These sub-files are, encoded into a set of video formats with different playback rates, supporting different media outlets.

In this system, the streaming engine incurs two alternative costs, demanding a trade-off in optimizing the total cost of service. First, it stores a partial set of transcoded content files in its local storage. This part of cost results from the rental of storage capacity from cloud service providers. Second, it transcodes the original content into the requested bitrate, resulting in a computing cost. It can be seen that these two parts of cost are competing with each other. On one hand, if the storage capacity is large, more transcoded content files can be stored, minimizing the opportunity of conducting real-time video transcoding. On the other hand, if more computing resources are provisioned, less transcoded contents files are needed to store statically in the local storage. In the following, we present our mathematical formulation into an optimization problem, aiming to minimize the total cost of service.

2.2. System Model

In this subsection, we present mathematical models for the cloud-based HTTP ABR streaming system, including content management model, user-request model and cost model.

2.2.1. Content Management Model

We assume that the system manages a set of M contents. For any video i ($i = 0, \cdots, M-1$), it is split into L_i segments, as illustrated in Fig. 2. We assume that L_i is bounded by above, i.e., $L_i \leq L$. Moreover, for each video i, it is transcoded into a set of N playback rates. For each playback rate of $r_{i,j}$ ($j = 0, \cdots, N-1$), a subset of content files are cached in the streaming engine. In view of the user viewing pattern, we assume that the first $n_{i,j}$ segments (i.e., from 0 to $n_{i,j} - 1$) are cached and the rest of $L_i - n_{i,j}$ segments (i.e., from $n_{i,j}$ to $L_i - 1$) are transcoded on live by the streaming engine.

2.2.2. User Request Model

It has been observed that not all viewers will complete the whole video clip, 60% of videos are watched for no more than 20% of their duration [6, 7]. As such, the user viewing pattern can be characterized by cumulative distribution function

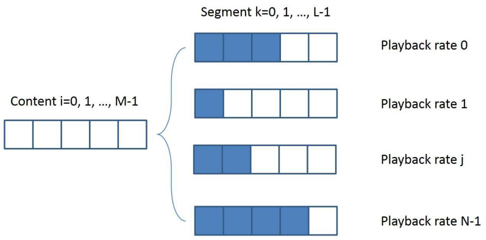


Fig. 2. Content management under the partial transcoding scheme. Shaded segments are cached, and unshaded segments are transcoded on live.

of $F_i(k)$. In particular, $F_i(k)$ denotes the probability in which the user would watch the video up to the kth segment. In practise, the viewing pattern can be approximated by a truncated exponential distribution, resulting the following formula,

$$F_i(k) = \frac{1}{K_i}(1 - e^{-\mu k}), 0 \le k \le L_i - 1, \tag{1}$$

where K_i is a factor for the ith content.

We consider a discrete time slot model and denote $x_{i,j,k}(t)$ as the number of users watching the kth segment of the ith content in the jth playback rate at time slot t. We assume we have the initial information at the beginning of the first time slot for how many users are watching a particular video segment (i.e., $x_{i,j,k}(0^-)$). Then, given the initial information, we have

$$\begin{aligned} x_{i,j,0}(0) &= x_{i,j,0}(0^-) + a_i(0)p_{i,j}, & (2)\\ x_{i,j,0}(t) &= a_i(t)p_{i,j}, & (3)\\ x_{i,j,k}(t) &= a_i(t)p_{i,j}\frac{1 - F_i(k)}{1 - F_i(0)}, & (4) \end{aligned}$$

where $a_i(t)$ is the number of arrivals requesting for the ith content in the first segment at time slot t, and $p_{i,j}$ is the probability of playback rate $r_{i,j}$ being requested[1], for which the condition of $\sum_{j=0}^{N-1} p_{i,j} = 1$ is satisfied. We assume $\mathbb{E}\{a_i\} = \lambda_i$ is known.

2.2.3. *Cost Model*

Two cost components incur in this proposed architecture, i.e., storage cost and computing cost.

Storage consumption: The storage consumption of the ith content is $B_i^S = \sum_{j=0}^{N-1} n_{i,j} f_i r_{i,j}$, where f_i denotes the scaling factor for video i and the file size is assumed to be proportional to its playback rate $r_{i,j}$. Thus, the total storage consumption to store all the contents is

$$B^S = \sum_{i=0}^{M-1} B_i^S = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} n_{i,j} f_i r_{i,j}. \tag{5}$$

Storage cost: It incurs storage cost in storing the cached video files. Specifically, for the ith content in the jth playback rate, the storage cost is $S_{i,j} = c_S f_i r_{i,j}$, where c_S denotes marginal price of storage space. Using this notation, the total storage cost for the ith content over time T is given by

$$C_i^S = \sum_{t=0}^{T-1} \alpha^t \sum_{j=0}^{N-1} n_{i,j} S_{i,j}, \tag{6}$$

where α is a discounted factor ($0 < \alpha < 1$). Thus, the total storage cost over time T is $C^S = \sum_{i=0}^{M-1} C_i^S$.

Computing consumption: The computing consumption of the ith content with playback rate of $r_{i,j}$ is $w_{i,j} = g_i r_{i,j}$, where g_i denotes the scaling factor for the ith content and the workload is assumed to be proportional to its playback rate. Using this notation, the total computing consumption for the ith content in the jth playback rate over time T is given by

$$B_{i,j}^W = \sum_{t=0}^{T-1} \alpha^t \sum_{k=n_{i,j}}^{L_i - 1} x_{i,j,k}(t) w_{i,j}. \tag{7}$$

We assume that the computing cost of a particular video is proportional to the number of user requests. Thus, the total computing consumption for the ith content over time T is

$$B_i^W = \sum_{t=0}^{T-1} \alpha^t \sum_{j=0}^{N-1} \sum_{k=n_{i,j}}^{L_i - 1} x_{i,j,k}(t) w_{i,j}. \tag{8}$$

The total computing consumption over time T is $B^W = \sum_{i=0}^{M-1} B_i^W$.

Computing cost: It incurs computing cost in real-time transcoding the video file into the requested playback rate. Specifically, the computing cost of transcoding the ith content into playback rate of $r_{i,j}$ is $W_{i,j} = c_W g_i r_{i,j}$, where c_W denotes marginal price of computing. Similar to the computing consumption, we can have the total computing cost for the ith content over time T, given by

$$C_i^W = \sum_{t=0}^{T-1} \alpha^t \sum_{j=0}^{N-1} \sum_{k=n_{i,j}}^{L_i - 1} x_{i,j,k}(t) W_{i,j}. \tag{9}$$

The total computing cost over time T is $C^W = \sum_{i=0}^{M-1} C_i^W$.

2.3. Problem Formulation

In this paper, we aim to minimize the time average storage cost and computing cost for partial transcoding. In a real system deployment, we observe that the storage space in the content cache can be filled quickly, in order to meet all the QoE requirement. In addition, the computation capacity of the system is also limited. Hence, the content management with partial transcoding scheme can be formulated as the following

[1] The requested playback rate is determined by the varying wireless channel. We assume we have the information of the probability distribution of users requesting different playback rates.

constrained optimization problem,

$$\min_{\vec{n}} \quad \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\{C^S + C^W\}, \tag{10}$$

$$\text{s.t.} \quad B^S \le \theta, \tag{11}$$

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\{B^W\} \le \rho, \tag{12}$$

where the expectation is taken over a_i. Eq. (11) represents the storage capacity constraint with total limited cache size θ and Eq. (12) represents the computing capacity constraint with workload threshold ρ.

3. SOLUTION OF COST OPTIMAL VIDEO TRANSCODING

In this section, we first make a transformation of the optimization problem and then solve it using Lagrange relaxation.

3.1. Transformation of the Optimization Problem

First, we note that from Eq. (2), the number of arrivals requesting for a specified segment changes in the first few time slots, due to the effect of the initial state $x_{i,j,0}(0^-)$. Suppose after T_s time slots, the system will enter into a stable state with a fixed average number of arrivals. The average number of arrivals requesting for a specified segment can be given as $E\{x_{i,j,k}(t)\} = \lambda_i p_{ij}\frac{1-F_i(k)}{1-F_i(0)}$. Then, considering Eq. (10), we notice that the cost incurred from 0 to $T_s - 1$, will become zero as $T \to \infty$. Therefore, it does not contribute to the overall average cost and the minimum average cost per time slot is determined by the cost incurred after T_s time slots. The same rationale can be adopted for the constraint Eq. (12).

Since request arriving distribution is the same for each time slot under the stability of the system, we make a one-shot policy for the cost-optimal real-time transcoding. The minimum average cost per time slot can be achieved by solving the following optimization problem,

$$\min_{\vec{n}} \quad \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[n_{i,j}c_S f_i r_{i,j} + \sum_{k=n_{i,j}}^{L_i-1} Q_{ij}(k)c_W g_i r_{i,j}] \tag{13}$$

$$\text{s.t.} \quad \pi_1 = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} n_{i,j} f_i r_{i,j} - \theta \le 0, \tag{14}$$

$$\pi_2 = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=n_{i,j}}^{L_i-1} Q_{ij}(k) g_i r_{i,j} - \beta \le 0, \tag{15}$$

where $Q_{ij}(k) = \lambda_i p_{ij}\frac{1-F_i(k)}{1-F_i(0)}, \beta = \frac{\rho(1-\alpha)}{\alpha^{T_s}}$. It's an integer linear program and known to be NP-hard, which can not be solved in polynomial time.

3.2. Approximate Solution of the Cost Optimal Video Transcoding

In this subsection, we rely on Lagrange relaxation to obtain the approximate solution to the optimization problem. We firstly rewrite the above optimization problem by introducing variables $m_{i,j,k}$, which denotes whether the kth segment of the ith video in the jth playback rate should be cached, and then relax the constraints (Eq.(14) and (15)) by bringing them into the objective function (Eq.(13)) with associated Lagrange Multipliers, we have

$$L(\vec{\mu}) = \min_{\vec{m}} \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=0}^{L_i-1}[A_{i,j,k}(\vec{\mu})m_{i,j,k} + B_{i,j,k}(\vec{\mu})] - \mu_1\theta - \mu_2\beta, \tag{16}$$

$$\text{s.t.} \quad m_{i,j,k} \in \{0, 1\},$$

where $\vec{\mu}$ is the Lagrange multiplier, and

$$A_{i,j,k}(\vec{\mu}) = [f_i(c_S + \mu_1) - Q_{ij}(k)g_i(c_W + \mu_2)]r_{i,j}$$

$$B_{i,j,k}(\vec{\mu}) = Q_{ij}(k)g_i r_{i,j}(c_W + \mu_2)$$

As we can see from Eq.(16), for a specified $\vec{\mu}$, $A_{i,j,k}(\vec{\mu})$ and $B_{i,j,k}(\vec{\mu})$ are constants. Then the subproblem is to minimize Eq.(16) by selecting appropriate $\vec{m}$. It can be easily solved by setting $m_{i,j,k}$ to 0 if the corresponding $A_{i,j,k}(\vec{\mu})$ is no less than zero, and setting $m_{i,j,k}$ to 1 otherwise. Thus the original problem is evolved to find the optimal $\vec{\mu}$ and then solve the subproblem. We use the subgradient method to obtain the optimal value of $\vec{\mu}$ in an iterative manner. The details of this algorithm are illustrated in Algorithm 1.

Algorithm 1 Subgradient Method for Partial Transcoding

```
Input:
    Initialize Scalar $\sigma \in (0, 2)$, $\vec{\mu} = \mathbf{0}$, s = 0,
Output:
repeat
    for all $i \in [0, M), j \in [0, N), k \in [0, L_i)$ do
        if $A_{i,j,k}(\vec{\mu}) >= 0$ then
            $m_{i,j,k} \leftarrow 0$
        else
            $m_{i,j,k} \leftarrow 1$
        end if
    end for
    Update upper bound ($UB$) and lower bound ($LB$) by calculating Eq.(13) and (16).
    Update $\pi_1$ and $\pi_2$ by calculating Eq. (14) and (15).
    Calculate step size:
        $\delta \leftarrow \frac{\sigma(UB-LB)}{\pi_1^2+\pi_2^2}, \vec{\mu} \leftarrow max(0, \vec{\mu} + \delta\vec{\pi})$
    $s \leftarrow s + 1$
until $s = 200$
```

4. NUMERICAL RESULTS

In this section, we first describe the experimental setting. We then analyze the performance of the approximate solution. Following that, we also compare the partial transcoding scheme with two alternative schemes.

4.1. Dataset Description and Experimental Setting

User Viewing Pattern: As mentioned in [6], video sessions have a very high probability to be aborted in a short time, mostly within 40 seconds. Similar observation was also demonstrated in [7, 8]. We use Eq.(1) to approximately fit the data in [6],

$$F_i(k) = \frac{1}{K_i}(1 - e^{-4.6\frac{k}{L_i - 1}}), 0 \leq k \leq L_i - 1, \tag{17}$$

where K_i equals to 0.98.

Video Popularity Distribution: Only 10% of popular videos account for nearly 80% user requests [5], most of videos are not frequently requested. We describe this characteristic by power-law distribution, $y \propto i^k$, where y is the PDF of a content viewed by i users in a time slot.

Storage and Computing Cost: We adopt the storage and transcoding pricing model from Amazon S3 Standard Storage and Elastic Transcoder. The storage price is $0.095/GB per month; the transcoding price is $0.015/minute for videos below 720p, and $0.030/minute for videos above 720p.

Video Duration and Bitrate: The average duration of online video is 4.3 minutes and only less than 5% of videos last more than 10 minutes [5]. For simplification, we assume video duration has a uniform distribution from 0 to 10 minutes, and each video content is transcoded into five kinds of bitrate files.

4.2. Comparison between Approximate Solution and Optimal Solution

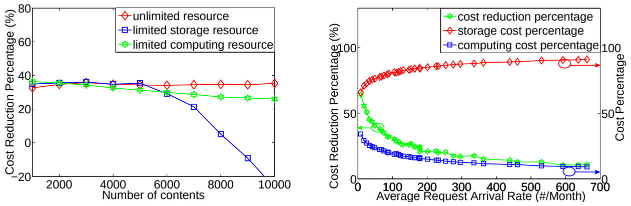
We compare its approximate solution with the optimal solution, we select the dimension of the variable $\vec{m}$ (i.e. the number of segments) from 10^2 to 10^7 and evaluate the approximation error ratio. As illustrated in Table 1, the approximate solution is close to the optimal solution and the error ratio does not increase as the dimension of $\vec{m}$ scales.

Table 1. Approximation Error of Different Dimensions

Dimension	10^2	10^3	10^4	10^5	10^6	10^7
Error Ratio(%)	0.0	2.4	2.6	2.8	2.7	2.7

4.3. Comparison with Alternative Schemes

We compare the performance of the proposed partial transcoding scheme with two alternative schemes, i.e., all-segments-stored scheme and full transcoding scheme without segmentation, given as follows:



(a) Varying content number (b) Varying request arrival rate

Fig. 3. Cost reduction over all-segments-stored scheme.

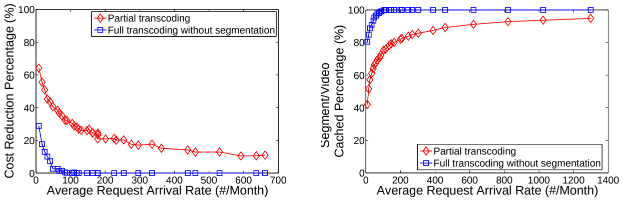
All-segments-stored scheme: it is a brute-force based scheme, which stores all the segments in different playback rates for each content. It would result in a large storage cost and zero computing cost, due to no transcoding operation.

Full transcoding scheme without segmentation: it is similar to the partial transcoding scheme that manages the tradeoff between storage cost and computing cost, but without file segmentation for content management. As such, the whole video file of a playback rate is either stored in local storage or transcoded online.

4.3.1. Comparison with all-segments-stored scheme

We plot the cost reduction percentage of partial transcoding scheme over the all-segments-stored scheme in Fig. 3. First, Fig. 3(a) shows the cases under varying content number and different resource constraints. The partial transcoding scheme can reduce the cost more than 30% when there are sufficient computing resource and storage resource (i.e., both Eq.(11) and Eq.(12) are inactive). While given limited computing resource, the partial transcoding scheme also has less total cost than the all-segments-stored scheme, but the cost reduction percentage drops slightly as the number of contents increases. However, when the available storage resource is limited (i.e., storage constraint is tight), the cost reduction percentage will decline dramatically as the number of contents increases. When there are more than 6000 contents, the constraint of Eq. (11) becomes active, indicating that the storage resource reaches its maximum utilization. Especially, when the number of contents is more than 8000, the partial transcoding scheme incurs more cost than the all-segments-stored scheme. This is because, with the increase of the number of contents, real-time transcoding will be conducted more frequently to satisfy the increasing number of user requests, due to insufficient storage space to cache segments. Therefore, one needs to strategically choose the storage capacity for designing an effective partial transcoding scheme.

Second, Fig. 3(b) shows the cases under varying average request arrival rate and a fixed number of contents. The partial transcoding scheme still outperforms the all-segments-stored scheme. As the contents' request arrival rate increases, however, the overall cost reduction percentage decreases. Particularly, the percentage of storage cost increases, while the



(a) Cost reduction percentage

(b) Cached percentage

Fig. 4. Comparison with full transcoding scheme without segmentation.

computing cost decreases. This is because, with the increase of contents' request arrival rate, there are more opportunity of transcoding segments that are not locally stored, resulting more computing cost. Thus, more segments will be cached in local storage rather than real-time transcoded with the increase of arrival rate.

4.3.2. Comparison with full transcoding scheme without segmentation

We compare partial transcoding scheme with full transcoding scheme without segmentation in Fig. 4. First, Fig. 4(a) plots the cost reduction percentage for these two schemes over all-segments-stored scheme. It shows that under different average request arrival rates, the partial transcoding scheme can have more cost reduction than the full transcoding scheme without segmentation. This is because, for the full transcoding scheme without segmentation, a video file has to be completely transcoded into the requested bitrate, even if a small fraction of the video is requested, resulting more computing cost. In addition, it can be observed that the full transcoding scheme without segmentation has restricted cost reduction. When the request arrival rate is low, it can still reduce the cost, but such a cost reduction disappears as the request arrival rate increases to around 100. In this case, it stores all the segments, without any cost reduction compared to the all-segments-stored scheme.

Second, Fig. 4(b) plots the cached percentage of these two schemes, respectively. As the average request arrival rate increases, the cached percentage of video files is much higher than the cached percentage of segments. Particularly, all the video files are stored under the full transcoding scheme without segmentation when the average request arrival rate reaches to around 100. This shows that the partial transcoding scheme is robust to the change of the request arrival rate and more adaptive than the full transcoding scheme without segmentation.

5. CONCLUSION

In this paper, we presented a mathematical model to achieve the cost optimal video transcoding by leveraging user viewing pattern. We first proposed a partial transcoding scheme for content management and formulated the minimization cost problem into an integer linear program. We then applied the Lagrange relaxation technique to obtain an approximate solution. Results show that our proposed method can save more than 30% of the operational cost. Moreover, compared with the method without partial transcoding scheme, our method is much more robust to the change of the request arrival rate. As future work, we will consider how to make caching policies in the case of the variation of video popularity.

6. REFERENCES

[1] Cisco, *Cisco Visual Networking Index: Forecast and Methodology, 2012 - 2017*, 2013.

[2] Weiwen Zhang, Yonggang Wen, Zhenzhong Chen, and Ashish Khisti, "Qoe-driven cache management for http adaptive bit rate (abr) streaming over wireless networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 1951–1956.

[3] Horacio Sanson, Luis Loyola, and Daniel Pereira, "Scalable distributed architecture for media transcoding," in *Algorithms and Architectures for Parallel Processing*, pp. 288–302. Springer, 2012.

[4] Ben Whitelaw, "Almost all youtube views come from just 30% of films," Tech. Rep., The Daily Telegraph, 2011.

[5] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.

[6] Alessandro Finamore, Marco Mellia, Maurizio M Munafò, Ruben Torres, and Sanjay G Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 345–360.

[7] Lucas CO Miranda, Rodrygo LT Santos, and Alberto HF Laender, "Characterizing video access patterns in mainstream media portals," in *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 1085–1092.

[8] S Shunmuga Krishnan and Ramesh K Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 211–224.