Poster: Popularity-Based Transcoding Workload Allocation for Improving Video Quality in Live Streaming Systems

Dayoung Lee
Department of Computer Engineering, Inha University
Incheon, Korea
22182181@inha.edu

Minseok Song
Department of Computer Engineering, Inha University
Incheon, Korea
mssong@inha.ac.kr

ABSTRACT

Transcoding is essential for live video streaming systems, but video streams in many channels may not be transcoded due to the processing capacity limitations of transcoding servers with multiple CPU nodes. To address this, we propose an algorithm that determines the transcodeable bitrate versions for each channel and allocates the transcoding tasks, by taking into account video quality, popularity, and workload balancing. Simulation results show that it can improve popularity-weighted video quality by up to 10.7% over other popularity-based alternatives.

CCS CONCEPTS

• **Information systems** → *Multimedia streaming*.

KEYWORDS

Transcoding, Live Streaming, Video Quality

1 INTRODUCTION

The demand for live streaming systems such as YouTube Live and Twitch.tv is growing rapidly. For example, one of the popular live streaming platforms, YouTube accounts for 35% of global mobile internet traffic [1]. They typically rely on DASH (dynamic adaptive streaming over HTTP) techniques to support various network conditions, requiring a significant transcoding server infrastructure to produce a number of bitrate versions for each channel.

Due to the limited processing capacity, transcoding services can only be provided to premium channels. Thus there are not enough video quality versions for all channels, so this makes it harder for the adaptation logic at the client to find a close match based on estimated bandwidth availability [3, 4]. In addition, live streaming requires real-time transcoding processing, so processing capacity needs to be continuously allocated to transcoding tasks for each channel [3]. To address this, we propose an algorithm that determines the transcodeable versions for each live channel and assigns each transcoding task to the processor node, with the aim of maximizing the popularity-weighted video quality.

CoNEXT '20, December 1–4, 2020, Barcelona, Spain
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7948-9/20/12…$15.00
https://doi.org/10.1145/3386367.3431665

2 ALGORITHM DESCRIPTION

Consider a live streaming system with N^{ch} channels. We introduce $V_{i,k}$, $(k = 1, ..., N^{\mathrm{br}})$ to express the k-th lowest bitrate version of channel i, where N^{br} is the number of total bitrate versions. Let $C^{\mathrm{ch}}_{i,k}$ be the processing rate in GHz required to transcode version $V_{i,k}$ to provide live transcoding, which can be obtained from measurements using programs such as Linux perf tool [3].

We assume that the lowest version for each channel is always transcoded to guarantee that in any case a version with a bitrate equal to or lower than the version requested by each viewer can be broadcast. Let S_i be a set of all the combinations for transcodeable versions for channel i, which are sorted in an ascending order of total CPU processing rates required to transcode all the versions in each combination, so that $S_i = \{\{V_{i,1}, V_{i,N^{\mathrm{br}}}\}, ..., \{V_{i,1}, V_{i,2}, ..., V_{i,N^{\mathrm{br}}}\}\}$. Let $S_{i,m}$ be the m-th element in S_i, $(m = 1, ..., N^{\mathrm{set}})$ where the number of elements in S_i is N^{set}, which is $2^{N^{\mathrm{br}}-2}$.

If $p_{i,k}$ is the access probability of version $V_{i,k}$, then $\sum_{i=1}^{N^{\mathrm{ch}}} \sum_{k=1}^{N^{\mathrm{br}}} p_{i,k} = 1$. Let $Q_{i,k}$ be the video quality index such as multi-method assessment fusion (VMAF) for version $V_{i,k}$. We introduce a new metric for version $V_{i,k}$, $G^{\mathrm{ver}}_{i,k}$ to represent popularity-weighted video quality (PWQ) when version $V_{i,k}$ is streamed as follows: $G^{\mathrm{ver}}_{i,k} = p_{i,k} Q_{i,k}$.

If version k is not in $S_{i,m}$, then the highest version lower than version k in $S_{i,m}$ needs to be delivered to avoid rebuffering due to network bandwidth shortage. We thus introduce $V^{\mathrm{high}}_{i,m,k}$ for $S_{i,m}$ to express the highest bitrate version index, which is lower or equal to the bitrate version k. Total PWQ for set $S_{i,m}$, $G^{\mathrm{set}}_{i,m}$ can be thus calculated as follows: $G^{\mathrm{set}}_{i,m} = \sum_{\forall V_{i,k} \in S_{i,m}} G^{\mathrm{ver}}_{i,V^{\mathrm{high}}_{i,m,k}}$.

A transcoding server has N^{node} processor nodes. Let C^{node}_j, $(j = 1, ..., N^{\mathrm{node}})$ be the total processing rate supported by node j. To maximize overall PWQ while meeting processing capacity at each node, the optimization problem determines the X_i-th combination for transcodeable versions in S_i and the $Y_{i,k}$-th processor node to transcode version $V_{i,k}$ as follows:

$$\text{Maximize} \quad \sum_{i=1}^{N^{\mathrm{ch}}} G^{\mathrm{set}}_{i,X_i}$$

$$\text{Subject to} \quad \forall j, (j = 1, ..., N^{\mathrm{node}}), \sum_{\forall V_{i,k}, Y_{i,k}=j} C^{\mathrm{ch}}_{i,k} \le C^{\mathrm{node}}_j.$$

This problem can correspond to a variant of a multiple-choice multiple knapsack problem, which is NP-hard. Since greedy methods based on profit to weight ratios perform well for knapsack problems, we develop a greedy heuristic algorithm as shown in Figure 1. Let X^{tmp}_i and $Y^{\mathrm{tmp}}_{i,k}$ be temporary variables for X_i and $Y_{i,k}$ during algorithm execution, respectively. All the values of X^{tmp}_i are

```
1: Temporary variables: $X_i^{\text{tmp}}$ and $Y_{i,k}^{\text{tmp}}$;
2: Boolean parameter: $flag \leftarrow TRUE$;
3: Array of parameters $I_{i,m}$'s: $A$;
4: for $i = 1$ to $N^{\text{ch}}$ do
5:     $X_i^{\text{tmp}} \leftarrow 1$;
6:     Find the node index $L$ with the largest remaining processing rate and allocate versions in $S_{i,1}$ to node $L$;
7: end for
8: while $flag$ do
9:     Find the maximum $I_{i,m} \in A$ for which $i = N$ and $m = H$, where $H > X_N^{\text{tmp}}$;
10:    for $\forall V_{i,k} \in S_{N,H}$ do
11:        Find the node index $L$ with the largest remaining processing rate when $X_N^{\text{tmp}}$ is updated by $H$;
12:        if Assigning $V_{i,k}$ to node $L$ violates the processing rate limit constraint then
13:            $flag \leftarrow FALSE$;
14:        end if
15:    end for
16:    if $flag = TRUE$ then
17:        $X_N^{\text{tmp}} \leftarrow H$;
18:        $Y_{i,k}^{\text{tmp}} \leftarrow L$;
19:    end if
20:    Remove $I_{N,H}$ from $A$;
21: end while
22: for $i = 1$ to $N^{\text{ch}}$ do
23:    for $\forall V_{i,k} \in S_{i,X_i^{\text{tmp}}}$ do
24:        $Y_{i,k} \leftarrow Y_{i,k}^{\text{tmp}}$;
25:    end for
26:    $X_i \leftarrow X_i^{\text{tmp}}$;
27: end for
```

Figure 1: Version set determination and allocation (VDA) algorithm.

initialized to 1 but can be increased to improve PWQ while satisfying processing capacity at each node. For this purpose, we define index parameters, $I_{i,m}$ for each selection for the m-th element in S_i, ($m = 2, ..., N^{\text{set}}$ and $i = 1, ..., N^{\text{ch}}$) as follows: $I_{i,m} = \frac{G_{i,m}^{\text{set}} - G_{i,1}^{\text{set}}}{\sum_{\forall V_{i,k} \in S_{i,m}} C_{i,k}^{\text{ch}}}$. The algorithm favors the selection with higher value of $I_{i,m}$ because this increases PWQ the most using the smallest possible CPU capacity. It thus finds the maximum $I_{i,m}$ for which $i = N$ and $m = H$, where $H > X_N^{\text{tmp}}$ from a set of $I_{i,m}$'s, updates X_N^{tmp} to H and allocates this transcoding task to the node with the largest processing rate if this would not violate processing capacity condition. This procedure is repeated until there exists no processing node to support the selected transcoding task. Finally, the values of X_i and $Y_{i,k}$ can be obtained.

3 SIMULATION RESULTS

We performed simulations to evaluate our scheme. We modeled processor nodes based on four recent commercial servers released between 2019 and 2020 [2]. The characteristics of one of these servers are then assigned to each node in sequence. We consider seven bitrates at four resolutions as recommended by public transcoding cloud provider, Zencoder [3]. VMAF values for each bitrate version were generated according to a normal distribution based on the measured values [4]. The access probability of each channel was modeled with the parameters of the gamma distribution observed on Twitch.tv [5]. We used measured parameters for processing rate (in GHz) required to transcode each bitrate version [3]. Relative popularity was set at 0.3, 0.2, 0.2, 0.1, 0.1, 0.05, and 0.05 to each bitrate version sorted from lowest to highest. N^{ch} was set to 105,000 [5].

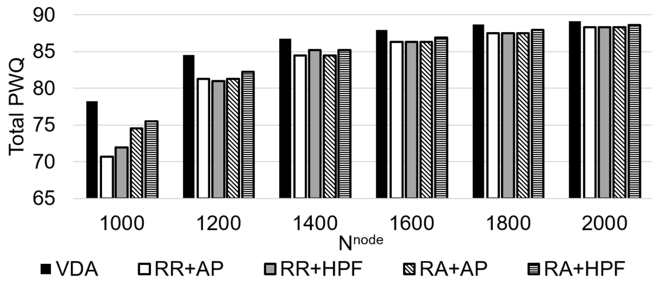

Figure 2: Total PWQ against N^{node}.

Two methods of task allocation can be considered: a round robin (RR) method of assigning each task to each node in sequence or a random (RA) method of assigning tasks randomly. Two popularity-aware methods can be considered for version determination: an AP (All versions for Popular channel) method choose $S_{i,N^{\text{set}}}$ for popular channels but $S_{i,1}$ for unpopular channels and a HPF (Highest Popularity First) method first transcodes the lowest bitrate versions and then transcodes the versions with the highest values of access probability successively. We compared our VDA algorithm with these four combinations: RR+AP, RR+HPF, RA+AP and RA+HPF.

Figure 2 shows how the total PWQ depends on the number of processor nodes. VDA always produces the highest total PWQ, exhibiting 0.62% and 10.67% higher than other popularity-based schemes. The PWQ difference is greater when the number of processor nodes is small, indicating that the VDA scheme is effective for heavy workloads.

4 CONCLUSIONS

We proposed a new algorithm that determines the transcodeable bitrate versions for each channel and assigns each transcoding task to an appropriate processor node in live streaming transcoding systems. Simulations showed that our scheme improves the overall PWQ compared to other popular-based alternatives, and this gap is pronounced when the number of processor nodes is small.

ACKNOWLEDGMENTS

This research was supported by Next-Generation Information Computing Development Program through National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2017M3C4A7080248), and partly by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2020R1F1A1068984). Minseok Song is a corresponding author.

REFERENCES

[1] https://www.sandvine.com/inthenews/youtube-accounts-for-35-percent-of-worldwide-mobile-internet-traffic.
[2] http://www.spec.org/power_ssj2008/results.
[3] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon. 2015. Transcoding live adaptive video streams at a massive scale in the cloud. In *Proceedings of the ACM Multimedia Systems Conference.* 49–60.
[4] Y. Qin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang, and C. Yue. 2019. Quality-aware Stategies for Optimizing ABR Video Streaming QoE and Reducing Data Usage. In *Proceedings of the ACM Multimedia Systems Conference.* 189–200.
[5] C. Zhang and J. Liu. 2015. On crowdsourced interactive live streaming: a Twitch.tv-based measurement study. In *Proceedings of the ACM Workshop on Network and Operating Systems Support for Digital Audio and VideoMarch.* 55–60.