An Encoding Method for Uniformalized VOD Quality in Cloud Transcoding System

Jeong-Mee Moo
Dept. of Media Tech
SK Broadband
Seoul, Korea
moonjmee@skcom

Jaeil Kim
Dept. of Media R&D
SK Telecom
Seoul, Korea
jaeilkim@skcom

Taeseung Hwang
Dept. of Media R&D
SK Telecom
Seoul, Korea
hwangtaeseung@skcom

Dongwon Kim
Dept. of Media R&D
SK Telecom
Seoul, Korea
dongwon.kim@skcom

Seongsooo Park
Dept. of Media R&D
SK Telecom
Seoul, Korea
seongsoo@skcom

Abstract— **To provide video on demand (VOD) services to customers on IPTV or OTT, it is necessary to transcode the contents provided by the content provider to suit the playback device and the user's viewing environment. In this paper, we propose a step-by-step quality control method to provide uniform video quality in transcoding IPTV VOD content in a transcoding system configured to fit a cloud computing environment. As a result of testing the performance by implementing it as a commercial service through parallel processing encoding workflow suitable for cloud environments, the proposed method meets the reference quality for each chunk section required to provide stable quality in various characteristic contents, and the encoding time is 1.21 times on average compared to a single stage of encoding.**

Keywords—Cloud Encoding System, Parallel Processing, Quality Based Encoding, *VOD* ***Transcoding***

I. INTRODUCTION

HTTP Adaptive Streaming (HAS) is a transmission method to provide stream optimized for various network states and the states of playback terminals, and the encoding server must generate a multi-rate stream encoded for multiple bitrates. Traditional HAS solutions use a fixed bitrate ladder, but this method has the disadvantage of not guaranteeing high quality for each content with various complexities and characteristics. Content with static characteristics may waste unnecessary resources due to fixed bitrate allocation, and content with fast movement, such as sports genres, may not be allocated enough bitrate, resulting in visual distortion. Recently, customized encoding methods for content have been proposed to improve this problem.

In Netflix [1] to [3], each content title was divided into a short interval unit, chunk, and the optimal bitrate ladder was generated by analyzing the bitrate-quality curve for the most complex chunk for each title. For this purpose, the Video Multi-method Assessment Fusion (VMAF) [4] quality metric was used. After encoding each chunk using a CRF parameter corresponding to the optimal bitrate, single-step or two-step VBR (Variable Bitrate) [5] encoding was performed according to the resulting bitrate. In the case of the Cock[3] method, generating bitrate ladder for each title takes a lot of time, and including the final VBR encoding step, at least two steps of encoding are required.

Lin[6]'s paper proposes to define the maximum bitrate and predict the constant rate-factor (CRF) parameter to perform stepwise encoding until the encoded bitrate does not exceed the target bitrate. If the final CRF value does not exceed the target bitrate, VBR encoding is performed using statistical information stored during CRF encoding, and if not, CRF parameters are predicted again. In this paper, we limited the CRF encoding step to a maximum of 4, and experimental results showed that an average of 2.52 steps were performed on CRF prediction. In this method, since stepwise encoding was performed under conditions not exceeding max bitrate, there is a problem that the result of encoding by the predicted CRF parameter does not always guarantee uniform quality, and an average of 3.52 steps of encoding is required including the final VBR encoding step.

Since many VOD contents of various genres are newly provided every day, rapid encoding and distribution workflows are required to effectively service them, and it is important to maintain uniform encoding quality for each scene and section to provide high quality to customers watching on TV.

In this paper, we propose a step-by-step quality control transcoding method to provide uniform quality in cloud-based IPTV VOD transcoding. For high transcoding performance, the flexible resource scalability provided by the cloud environment was utilized to separate each content into short chunk units to be parallel transcoded. In addition, instead of analyzing complexity by input content, encoding statistics are used to predict optimal CRF parameters for each content genre and series, and encoding is performed to satisfy the target quality of each chunk through stepwise encoding according to the encoding quality by chunk interval. In addition, the maximum bitrate is used as an additional parameter for CRF encoding for transmission bandwidth constraints to encode the bitrate so that it does not exceed the maximum value even in complex intervals. Section II describes the system configuration for parallel processing of video content using the cloud environment, Section III describes the proposed step-by-step encoding scheme, Section IV describes the transcoding results for the proposed scheme in a cloud-based transcoding system, and Section V concludes.

II. CLOUD-BASED PARALLEL VIDEO TRANSCODING

The most effective way to utilize the characteristics of cloud computing in transcoding systems is to process video content in parallel. In a single computing environment, the performance of the CPU or GPU determines the performance of transcoding, but in a cloud environment, the construction of parallel processing workflows that segment, transcode, and merge images becomes more important.

A. *Workflow design for parallel processing*

One of the advantages is that large-scale computing resources can be used simultaneously in a cloud environment. To effectively utilize this in video content transcoding, it is necessary to design a workflow that can divide the encoded content into small chunks, transcode these chunks independently and then merge the results again. Fig. 1 shows the configuration of a workflow for parallel processing of VOD content.

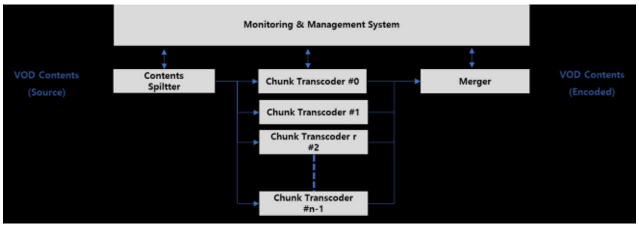


Fig. 1. The configuration of a workflow for parallel processing

As shown in Fig. 1, the content requiring encoding is divided into chunks by the Splitter module and provided as input from individual chunk transcoders. Each chunk transcoder receives the parameters required for encoding from the management module and performs the transcoding operation. The monitoring module monitors the status of individual transcoder operations, and when all the chunk transcoder operations are completed, it instructs the merge module to integrate the transcoded files. The Merger module merges the results transcoded by each Chunk transcoder in order and generates a single VOD file. For this workflow to be possible, integrity must be guaranteed in the process of zero loss segmentation of video content, accurate encoding of segmented content, and integration of transcoded result files.

B. *Implementing parallel processing workflow*

To divide video content into chunks and effectively perform parallel processing, accurate file division must be performed in the splitter module. Splitter analyzes the input source video file to obtain information such as file size, frame rate, bit rate, codec type, and total playback time. In addition, the file is divided into chunks of a fixed time so that it can be processed by transcoder. In this case, each chunk should be allowed to overlap a certain interval to prevent video image loss or audio signal loss. This is to allow images and audio to be accurately decoded and transcoded at a predetermined time in the video transcoding process.

The chunk transcoder operates independently and transcodes video content in a specified interval. The structure of the chunk transcoder is shown in Fig. 2.

As shown in Fig. 2, the chunk transcoder consists of a decoder, an encoder, and a VMAF module. The decoder receives a video chunk including an overlap and allows the video to be accurately recovered in the target transcoding interval. For the recovered video, the encoder converts the image into a video format to be converted by referring to the specified parameter. It passes through the converted video VMAF module and checks whether it has been converted to meet a set quality standard. If the encoded video does not meet the quality standard, encoding is performed again with higher quality through parameter adjustment.

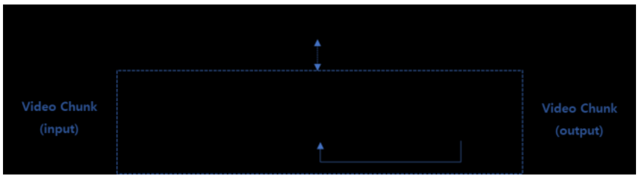


Fig. 2. The structure of the chunk transcoder

Merger performs the task of re-merging the completed video chunks from each chunk transcoder. At this time, since the task completion time of each chunk transcoder may be different, the video chunks are sequentially sorted and merged through a sufficient buffering process.

III. PROPOSED VOD ENCODING METHOD

In this paper, we propose CRF encoding for target quality to maintain consistent quality for each chunk. Instead of omitting VBR encoding in Encoding workflow, the target max bitrate value was entered as encoding option to limit the bitrate, and Netflix's VMAF [4] was used as the quality metric. Instead of analyzing CRF-Quality by input content title for fast encoding workflow, encoding result statistics of encoded content were used to predict CRF parameters. The CRF encoding step was limited to a maximum of three times, and the encoding quality of each step was compared with the target quality, and if this was not satisfied, re-encoding was performed in the next step. CRF parameter prediction methods for encoding each step are described in III.A to III.C.

A. *Encoding using genre CRF statistics (Stage 1)*

VOD, currently provided on IPTV or OTT, consists of various genres and series. Genres are divided into sports, dramas, movies, and animations, baseball, soccer, and golf games within the sports genre, and CSI and Game of Thrones within the drama genre are divided into series and managed. This paper assumes that content titles have similar characteristics within the same series of the same genre and updates the statistics by genre and series of encoded results to apply the optimal CRF statistics of the series as initial CRF parameters during new encoding. In equation (1), $CRF_i^{(i)}$ and

CRF'$_{g,k}$ means the mean CRF parameter values for the first stage CRF parameter of the ith chunk and the series ID k of the genre g, respectively.

$$CRF_i^{(1)} = CRF'_{g,k}, \quad \text{where } g \text{ and } k \text{ are genre and series id.} \tag{1}$$

Using the first stage CRF parameter CRF$_i^{(i)}$ obtained from the above equation and the predefined maximum bitrate (B_{max}), the encoding process ends if the quality value Q(CRF$_i^{(i)}$) measured after the first-stage encoding of the chunk i is greater than the target Q_{target}, otherwise the second-stage encoding is performed

B. Encoding using CRF-Quality statistics by genre(Stage 2)

If the first-stage encoding result of chunk i does not meet the target quality, we predict CRF$_i^{(2)}$ for second-stage encoding using a genre-specific CRF-Quality graph obtained by encoding with all CRF parameters in the available range for 200+ chunks in advance. Fig. 3 is a schematic diagram of various CRF values and corresponding encoding quality (Quality), i.e., VMAF values, for a drama genre with 1080p resolution. resolution. Even within one same genre, the characteristics of the complexity of the content were classified into four, allowing different quality values to be predicted according to the complexity of the encoded content.

As shown in the example in Fig. 3, if the first stage CRF parameter and quality measurements of the drama genre were 26 and 91.5 respectively, the CRF-specific quality score for the new encoding result can be predicted in proportion to the distance between the corresponding quality measurement score and the quality score of the two adjacent graphs.

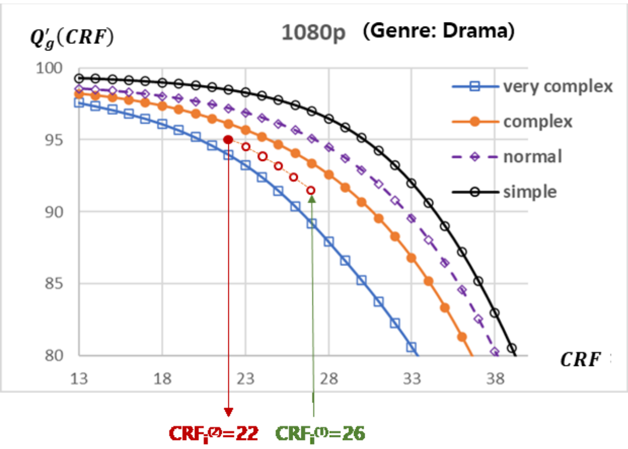


Fig. 3. CRF-Quality curve based on the statistics of 1080p drama genre

In equation (2), Q'$_g$^{lower}*(c)* and Q'$_g$^{upper}*(c)* represent two adjacent statistical quality values for each CRF value c, and $\hat{Q}(c)$ represents the predicted quality value. The maximum CRF value, in which the predicted quality value exceeds the target quality, was used as a two-step encoding parameter.

$$CRF_i^{(2)} = argmax(\hat{Q}(c) > Q_{target}), \ (CRF_{min} \le c \le CRF_i^{(1)}), \tag{2}$$

$$\text{where } \hat{Q}(c) = Q'_{g^{lower}}(c) \times \alpha + Q'_{g^{upper}}(c) \times (1-\alpha).$$

If the quality measurement score($Q(CRF_i^{(2)})$) satisfies the target quality after two-stage encoding, the encoding process is terminated; otherwise, three-stage encoding is performed.

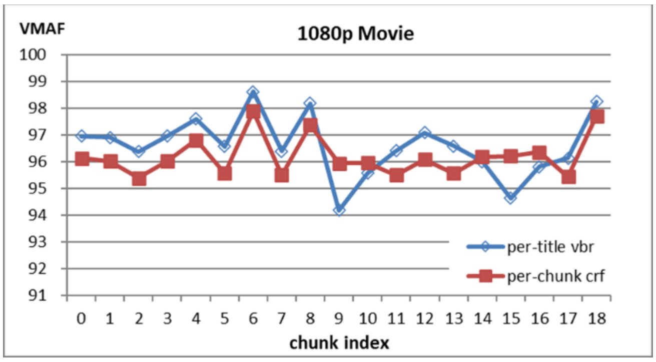
C. Prediction encoding using previous step results (3 Stage 3)

If the target quality was not met in the second stage, the characteristics of the chunk were judged to be outside the general statistical characteristics, and the three-stage CRF parameter was predicted using the first and second stage encoding results of the chunk, not statistical information as shown in equation (3).

$$CRF_i^{(3)} = CRF_i^{(2)} + \frac{Q_{target} - Q\left(CRF_i^{(2)}\right)}{Q\left(CRF_i^{(2)}\right) - Q\left(CRF_i^{(1)}\right)} \times \left(CRF_i^{(2)} - CRF_i^{(1)}\right) \tag{3}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the performance of the step-by-step quality control encoding scheme proposed in this paper, the results of the existing commercial encoding scheme and the average VMAF score for each chunk interval of the proposed scheme are compared in Fig. 4.



(a) Conventional per-title VBR encoding of on-premise encoder: 6,000kbps, VMAFmean: 96.81, VMAFs: 3.39

(b) Proposed cloud based per-chunk CRF encoding: 4500kbps, VMAFmean: 96.44, VMAFs: 3.21

Fig, 4. Comparison of chunk VMAFs between conventional per-title and proposed per-chunk encoding.

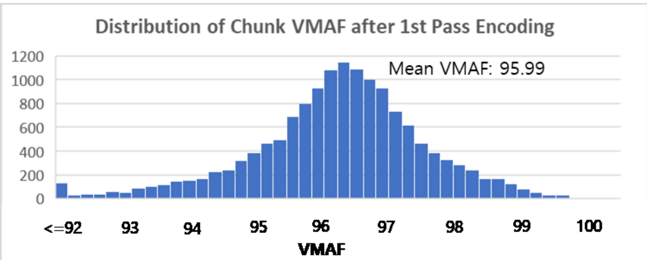
The content used in the comparison was a movie genre with a resolution of 1080p, and the chunk of the proposed method was applied in 5 minutes length. Conventional commercial encoding methods measure average VMAF scores in 5-minute increments for comparison with two-stage VBR encoding methods for the Per-title target bitrate. In Fig. 4, the proposed method shows that the VMAF standard deviation decreased from 3.39 to 3.21 compared to the existing commercial method, and that all VMAF values exceed the target 95 points. In addition, the average bit rate for content also decreases from 6,000 kbps to 4,500 kbps because the target quality is satisfied even with the low bit rate in the high-quality section.

In this paper, for a comparison of the step-by-step encoding performance in the proposed method, the results of each encoding step-by-step for 2,230 contents (total 14,700 chunks) of various genres used in commercial services are shown in Table 1. In Table 1, 12,032 chunks, approximately 81.85% of the total chunk count, met the target quality in the first stage encoding, and approximately 14.44% and approximately 3.71% met the target quality in the second and third stage encoding, respectively.

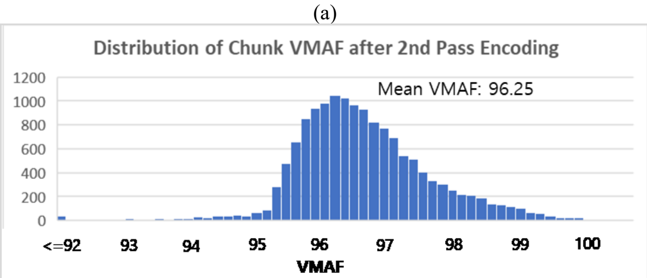
TABLE I. TABLE 1. AVERAGE CHUNK VMAF FOR EACH ENCODING PASS

Num Chunk	Percentage (%)	1st pass	2nd pass	3rd pass	Final VMAF
12,032	81.85	96.44			96.44
2,122	14.44	94.22	95.73		95.73
546	3.71	92.97	93.97	95.09	95.09
14,700	100				96.29

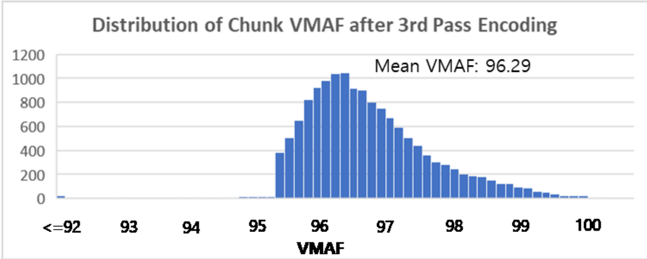
In addition, as shown in Table Ⅰ, the number of chunks that performed this two-step or higher encoding was 2,668, which is about 21.86% of the total chunk number, and the overall encoding time was about 1.21 times that of single-step CRF encoding. This means that the encoding time will be 0.79 times less than the conventional two-step VBR, which can provide content quality above the equivalent level. Fig. 5 shows the VMAF distribution of the entire chunk at the completion of each encoding step using the encoding results in Table Ⅰ. From this distribution, chunks that had a lower VMAF value than the target quality VMAF 95 points after encoding the first stage have a value of 95 points or more through the re-encoding process.



(a)



(b)



(c)

(a) VMAF distribution after 1st pass encoding
(b) VMAF distribution after 2nd pass encoding
(c) VMAF distribution after 3rd pass encoding
Fig. 5. Distribution of chunk VMAF after each encoding pass

V. CONCLUSIONS

In this paper, we propose a cloud-based chunk-based step-by-step quality management using CRF encoding method for delivering uniform quality VOD services. We evaluate the proposed approach in a production environment by implementing a transcoding system that performs parallel processing on a chunk level and a system that allows quality measurement on the transcoded results at the chunk level.

The performance evaluation results on a commercial system demonstrate that the proposed approach enables the delivery of uniform quality VOD content services with efficient bitrate allocation compared to existing encoding approaches, with only an additional processing time of 1.21 times the processing time of single-step CRF encoding. In practice, the proposed method will enable more efficient transcoding services for HLS Adaptive Streaming in transmission environments requiring various bitrates and resolutions.

ACKNOWLEDGMENT

This paper was written by results of a collaboration project between SK telecom and SK broadband.

REFERENCES

[1] A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, "Per-Title Encode Optimization," https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2.

[2] I. Katsavounidis, "Dynamic optimizer - a perceptual video encoding optimization framework," https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.

[3] J De Cock, Z Li, M Manohara, A Aaron, "Complexity-based consistent-quality encoding in the cloud," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1484-1488.

[4] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, 2016.

[5] L. Merritt and R. Vanam, "Improved rate control and motion estimation for h.264 encoder," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2007, pp. 309–312.

[6] Y.-C. Lin, H. Denman, and A. Kokaram, "Multipass encoding for reducing pulsing artifacts in cloud based video transcoding," *in IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 907–911.