The UHD content packet format for distributed transcoding

YoSeph Choi[1], Seongsin Jeon[2] and WooChool Park[3]
Contents Convergence Research Center, Korea Electronics Technology Institute
11, World cup buk-ro 54-gil, Mapo-gu, Seoul, 03924, Korea
E-mail: [1]yschoi@keti.re.kr, [2]seongsin@keti.re.kr, [3]wcpark@keti.re.kr

Abstract: The extra high-efficient codec technology that based on the HEVC is essential for streaming the UHD content over the Internet. It requires an extra high-efficiency transcoder technology to simultaneously support various resolutions because it expected to have SD, HD, UHD content consumption environments coexist in the N-screen. Therefore, from the existing low-efficiency compression format (H.264 / AVC, VC - 1, such as ProRes) to the extra high-efficiency multi- resolution images that is convertible to (SD, HD, UHD) compression format from the transcoder described is essential. At domestic and abroad are currently in progress to develop technology for real-time transcoding UHD, however the UHD Ultra-Capacity Due to the characteristics of the situations that require research and development of cloud-based distributed transcoder technology. In this paper, we propose a Packet Analyzer and KMSP packet structure to perform effectively distributed transcoding.

Keywords—**UHD, Streaming, HEVC, Transcoding**

1. Introduction

Transcoding refers to a technique for encoding to convert the input to any of the video content to the desired codec and format profiles. The UHD-class high-quality content is needed because many operations in the coding capacity of at least 23MB per frame invitation. The high-quality contents of the UHD-class shorten the processing time for coding because of distributed processing systems it is possible to divide a single business process and at the same time.

The distributed transcoding system analyzing the packets of the input compressed video and after splitting the stream of content at a predetermined time interval and a function for distributed transcoding the respective divided data. The distributed transcoding system, it is important to partition and format the definition of efficient business division data because it looks for a performance difference depending on the shape of the segmented work.

In this paper, we introduce the Hadoop-based distributed transcoding system and the Packet Analyzer that analyzing the input image to perform the transcoding effectively and the KMSP packet format for data partitioning.

2. The Distributed Transcoding System

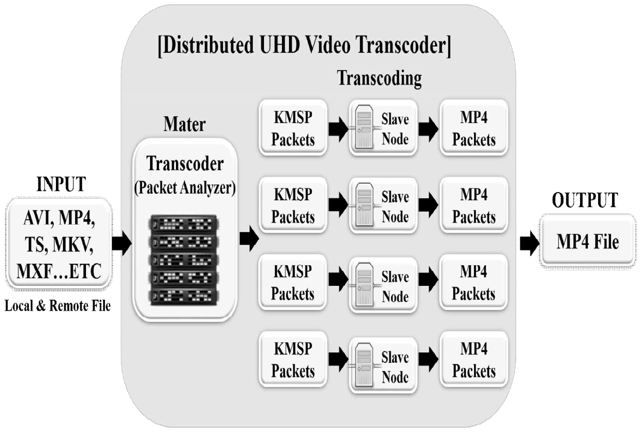

Figure 1. The Architecture of Distributed Transcoding System

Distributed transcoding system to mix the newly after analyzing the packets of the compressed input image and dividing the stream at a predetermined time interval by the parallel distributed transcoding compressed video stream and an audio stream functions to create a new image format.

The overall configuration of a distributed transcoding system also is configured the Master Transcoder (Packet Analyzer) for analyzing the input image and an input image is divided into predetermined time intervals the same as Figure 1 in Slave Nodes balancing transcoding. The KMSP packet is a packet of KETI Multi - Stream Protocol format proposed in order to effectively divide the image.

3. Video Packet Analyzer of UHD Distributed Transcoding

3. 1 The Structrue of Packet Analyzer

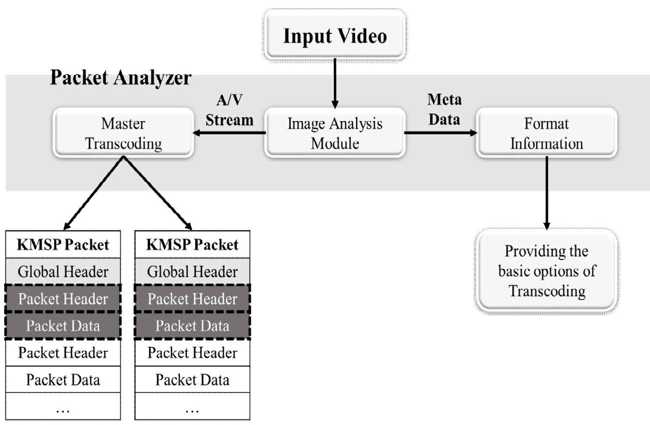

Figure 2. The Structrue of Packet Analyzer

The Figure 2. shows the structure of Packet Analyzer. Packet Analyzer is responsible for analyzing and dividing the video packets of different formats, like the Figure 2. when the video image is the input image and to identify the information of the speech codec a certain time a Stream of the input video data is divided and stored in a (for example , 5 seconds) unit. At this time, stored as KMSP Packet structure for distributed transcoding.

3. 2 The Function of Packet Analyzer

The Packet Analyzer is divided into two function module for dividing the time of the image analysis module and the images .

3. 2. 1 Image Analysis Module

The Image Analysis Module analyzes the image in various formats and the image and extracts image information format. Also, the Image Analysis Module provides a default option information for transcoding using the open source FFMPEG library. In addition, The Image Analysis Module delivers the video format information even to the image time division module.

3. 2. 2 Video time-division module

The Video time-division module divides the input picture at a constant time unit module to generate a packet KMSP. The Video time-division module performs upload the generated KMSP packets to the distributed file system, HDFS (Hadoop Distributed File System).

3. 3 The Packet Format of KMSP

A data packet of the divided content generation from the Packet Analyzer should include the basic options information and video and audio format information for the partition information, the transcoding. The split structure of each of the content data packet is shown in Figure 3.

The packet data of the divided content is composed of video and Global Header, Audio Packet. The Global Header is composed of the division information to the output stream Header Profile. The Profile Number of Header is equal to the number of output streams. Also, the video, the audio packet is composed of a Packet Data Header is contained in the packet type and the timestamp information and containing the actual video and audio data is Data Packet Payload

Global Header , Profile Header , Packet Header AV is expressed in terms of Figures 4, 5, 6 to C structure .

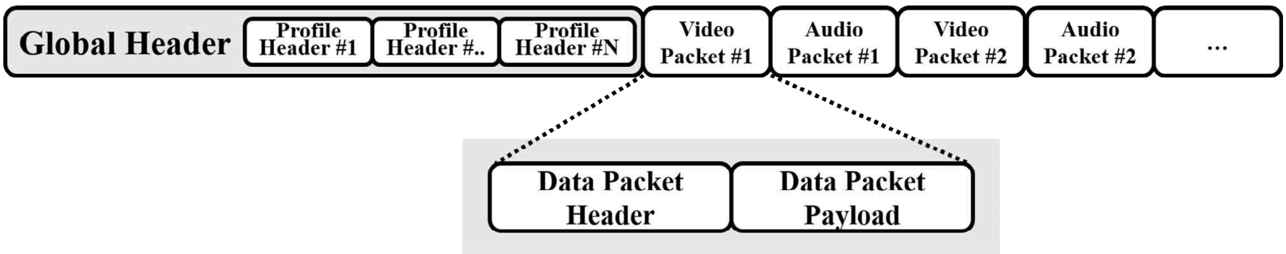

Figure 3. The Structure of the KMSP packet

```
struct sKmspGlobalHeader
{
        uint32_t nSize;
        uint32_t uiSeqId;
        uint32_t uiStartFrameNum;
        uint32_t uiEndFrameNum;

        // video input part
        uint32_t uiVideoCodecID;
        int32_t iVtimeBaseNum;
        int32_t iVtimeBaseDen;
        uint32_t uiPicWidthOrg;
        uint32_t uiPicHeightOrg;
        uint32_t uiPixelFormat;

        // audio input part
        uint32_t uiAudioCodecID;
        int32_t iAtimeBaseNum;
```

```
int32_t iAtimeBaseDen;
            uint32_t uiNumChannels;
            uint32_t uiSampleRate;
            uint32_t uiChannelLayout;
            uint32_t uiSampleBitDepth;
            uint32_t uiSampleFormat;

            // A/V output part
            uint32_t uiNumOutputProfiles;
            uint16_t uiExtraVHeaderLen;
            uint16_t uiExtraAHeaderLen;
            uint8_t* pExtraVHeaderBuf[uiExtraVHeaderLen]
            uint8_t* pExtraAHeaderBuf[uiExtraAHeaderLen]
            sProfileHeader StreamInfo[uiNumOutputProfiles]
};
```

Figure 4. The Format of Global Header

```
struct sProfileHeader
{
            // total output part (A/V/F)
            uint32_t uiBitrate;

            // video output part
            uint32_t uiPicWidth;
            uint32_t uiPicHeight;
            uint32_t uiFrameRate;
            uint16_t uiChromaFormat;
            uint16_t uiBitDepthV;
            uint32_t uiBitrateV;

            // audio output part
            uint32_t uiAudioCodecIDout;
            uint32_t uiSampleRate;
            uint16_t uiNumChannels;
            uint16_t uiBitDepthA;
            uint32_t uiBitrateA;
            uint32_t uiChannelLayout;
};
```

Figure 5. The structure of the Profile Header

```
struct sDataPacketHeader
{
            uint32_t uiFrameFlags;
            uint32_t nPacketSize;
            int64_t   nPTS;
            int64_t   nDTS;
            uint8_t
payload[nPacketSize];
};
```

Figure 6. The Structrue of the Packet Header

4. Performance Analysis

In this paper, the packet format for the division of the technology and the data for analyzing the packets of compressed video format in order to effectively perform dispersion UHD transcoding of content. We built the Hadoop based distributed transcoding system and the test was as shown in Table 1.

Table 1. System Environment

OS	Windows 7 SP1 64bit
CPU	Intel Xeon E5-2687W v2 (8-core, 3.4GHz)
RAM	32GB
Hadoop Cluster	Master1, Slave7
Contents	4K(3840x2160) MP4 (H.264/AVC)

After a progress transcoding using the content set forth in Table 2 were obtained the results shown in Table 3. It is expected to more sophisticated techniques for distributed transcoding of the next UHD contents.

Table 2. 4K UHD Contents

Contents Name	Total Frame	Bitrate(Kbps)
Jack	3877	43584
City	2706	39215
Sport	4286	41380
Nature	6237	40910
YoungDay	6963	40660
FIFA	53949	24501
Average	13003	38375

Table 3. 4K UHD Video Transcoding Speed, Quality and Compression Ratios

Contents Name	Speed(fps)	Output Bitrate	Output Video Quality (dB)
Jack	33.80	3503.88	44.94
City	31.58	3219.36	40.61

Sport	28.69	7759.64	40.22
Nature	44.26	5921.36	38.36
YoungDay	44.84	1653.70	51.80
FIFA	29.44	9403.10	40.58
Average	35.44	5243.51	42.75

5. Conclusion

The HEVC codec technology based on ultra high efficiency is essential for UHD contents to stream on the Internet. Therefore it expected to have SD, HD, UHD content consumption environment coexist in the N-screen service.

It is necessary to simultaneously support different resolutions. Therefore, It is essential transcoder from the existing low-efficiency compression format(H.264/AVC, VC-1, such as ProRes) to high-efficiency multi-resolution(SD, HD, UHD). Although the ongoing globally development of the technology for real-time transcoding UHD, UHD Content is ultra-capacity owing to the characteristics of the situations that require research and development of cloud-based distributed transcoder technology. In this paper, We propose packet analyzer and KMSP packet format in order to perform effectively a distributed cloud transcoding.

ACKNOWLEDGMENT

This work was supported in part by the IT R&D program of Ministry of Science, ICT and Future Planning, Republic of Korea under Grant No. 1711021671

6. References

[1] Sullivan, Gary J., et al. "*Overview of the high efficiency video coding (HEVC) standard*." Circuits and Systems for Video Technology, IEEE Transactions on 22.12 (2012): 1649-1668.

[2] "Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding," ISO/IEC 23008-2, 2013

[3]http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware

[4] x265 project, http://x265.org/

[5] Chen, Keji, et al. "Efficient SIMD optimization of HEVC encoder over X86 processors." Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific. IEEE, 2012.

[6] Wang, Xiangwen, et al. "*Paralleling variable block size motion estimation of HEVC on CPU plus GPU platform*." Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on. IEEE, 2013.

[7] https://tech.ebu.ch/testsequences/

[8] https://tech.ebu.ch/docs/hdtv/svt-multiformat-conditionsv10.pdf

[9] Bjontegard, Gisle. "*Calculation of average PSNR differences between RD-curves*." ITU-T VCEG-M33 (2001).