

The Research of video format conversion based on educational video-on-demand

Fei Wan^{1, a}, Yi Dai^{2, b}

¹ Zhuhai Radio & TV University, Zhuhai, China

² Corresponding author, South China Normal University, School of Information Technology In Education, Guangzhou, China

^axiner198509@qq.com, ^bzhsjyj@qq.com

Keywords: educational video-on-demand; video transcoding; FFmpeg ; MEncoder

Abstract. With the popularity of the Internet, User Generated Content increasingly attracts more attention, which proposed the new requirements of the educational video-on-demand, providing the function of video upload for educators and educated. However, there is now a greater variety of video, and the video data which has being digitized is very large. So how to effectively compress video data & audio data, and stores it into the appropriate format to become a serious problem of the Server. FFmpeg and MEncoder, the two kinds of open-source & cross-platform command line program for video compression and processing, has become the most popular core of video compression and processing software. In this paper, the difference between FFmpeg and MEncoder has been tested, based on a combination of both advantages and video coding information. We design the video trans coding module of Server, providing a unified server-side video file format Technology solutions for developers.

Introduction

The VOD (video on demand) system began to rise with the violent development of Computer network technology and multimedia applications. It has gradually become an important means for people to obtain network information and played an important part in distance education^[1]. The VOD system, which is audio-visual, interactive and meets the user's needs, has a good application to education. It not only provides synchronized audio, video and other multimedia information for the educated in real-time; at the same time, to a certain extent, it has realized the demand of education and has broken the traditional education limitations in time and space, so that the educated can choose and obtain all kinds of interesting multimedia information according to their own needs^[2].

Educational Video on Demand(EVOD) system, which means the VOD system's application in education, is the same with the common VOD system in its structure and data transmission mode. But materials in the EVOD system should be scientific and educational, which is different with the common VOD system. Because it is mainly used in students' autonomous learning, students' extracurricular knowledge development, classroom teaching and teacher professional development^[3]. Therefore, the development of EVOD system relies heavily on progress of VOD technology. The new request to the EVOD system in respects of user participation and construction has been put forward because of: the rapid development of Web 2.0, the access speeding of mobile network, and the gradually improving terminal processing ability.

Challenge of EVOD system

The popularity of the Internet gave birth to the concept of user generated content (UGC), which means the user could share their original content to other users through the Internet platform. The way of surfing the Internet has been changed from download to both for download and upload. YouTube, MySpace, Youku and other mainstream VOD sites, have the function to support uploading and sharing the original video. The successful application of UGC in VOD system puts forward the new request to

the EVOD system. This means that both educators and students could send video in the EVOD system and the system could automatically share and maintain these resources so that it's easy and fast to send the educational resources. This helps to strengthen the communication between teachers and students, students and students, and could further improve the students' learning effects and teachers' teaching level.

The current open oriented EVOD system at the same time is also facing a problem. A large number of different formats of video information are intertwined with many video coding standards because of the video coding information diversification (format, resolution, codec, etc) ^[4]. Though it could be solved by limiting the upload video format, it may increase the burden of the user. It has become a very important problem of the EVOD system that how to effectively compress the video and audio data and store them in an appropriate format. Unified video file format, which is conducive to the browser compatibility, could reduce use cost for the educated, enhance the interface-friendly, easily manage the site server convenient, and then the video material is easy to be applied to teaching, which has important significance^[5]. Aiming at solving this problem, the author mainly researches on the two video format conversion technology: FFmpeg and MEncoder, and provides unified video file format technical solutions for EVOD system developers.

Video format conversion technology

The most popular video compression and processing software core are FFmpeg and MEncoder. Both are open video compression processing command line program source cross platform. Many FFmpeg developers are also the developers of Mplayer, so the two are closely related. In addition, FFmpeg and MEncoder have been widely used in most of the video player software, such as: Storm video player using the FFmpeg, Mplayer player using MEncoder.

FFmpeg

FFmpeg is now very widely used free open source projects, which provides the complete solution for recording audio and video, conversion, encoding and decoding. It contains libavcodec, which is the current leading audio and video codec library and redevelops many codec in the libavcodec library, in order to ensure the quality and performance. FFmpeg which is based on the Linux platform, can be compiled and run on Windows and other mainstream operating system. A lot of good players, such as storm video, TCPMP, VLC, are using the FFmpeg decoder.

Use of FFmpeg

The basic syntax of FFmpeg is : `ffmpeg [inputfile_options] -i inputfile [outputfile_options] outputfile`, where "inputfile_options" is the source file format options , "-i" means reading the source file command , "inputfile" is the name of the source video files with full path (the file in hard disk, network flow, audio and video capture, etc) , "outputfile_options" is the output file format options , "outputfile" is the output file name with full path. The common options of FFmpeg are shown in table 1.

Table 1 Common Options of FFmpeg

Options	Function Description
-f	Compulsory input or output formula
-c/codec	Select codec for the source file or output file
-b	Set bit flow
-vframes/-frames:v	Set the video frames of the output file
-r	Set the frames
-s	Set the frame size, format: w*h(width*height)
-aspect	Set the aspect ratio of the video display, such as:4:3, 1.7777
-vcodec/-codec:v	Set the video codec
-ar	Set the audio sample rate
-aq	Set the audio quality
-ac	Set the audio channel
-acodec/-codec:a	Set the audio codec
-sample_fmt	Set the audio sample format

FFmpeg video format conversion technology

The FFmpeg command line could help the video format conversion and video images extraction. Video format conversion command, as shown in Figure 1, can convert “input.avi” files to “output.flv” files, set the output file to use the flv video encoder, set frame rate to 24fps, the video coding rate to 500bps, the resolution to 720 x 480, the output file to use the MP3 audio encoder, the audio sampling rate to 22050Hz, and the audio coding rate to 56kbps.

```
ffmpeg -i input.avi -f flv \
-vcodec flv -r 24 -b 500 -s 720*480 \
-acodec mp3 -ar 22050 -ab 56 output.flv
```

Figure 1 FFmpeg format conversion command

The FFmpeg video images extraction command , as shown in Figure 2, could extract one input.avi video frame every second and name the images as ”output-001.jpg”, “output-002.jpg”, etc. The resolution of the picture will be reset according to the "W * H" value. To extract a finite number of frames, you can add a -vframes or -t option; you can also add the -ss option, which starts to extract images from a particular point.

```
ffmpeg -i input.avi -r 1 -s WxH -f image2 output-%03d.jpeg
```

Figure 2 FFmpeg video images extraction command

MEncoder

MEncoder is the coding tool that comes with Mplayer, a Linux video playback tool. It is an efficient video compression and format conversion system using C language. MEncoder, based on the command line, needs to manually add the required parameters by using the command line. It supports a variety of video formats, optimizes compilation in view of different CPU instruction set optimized compiler, and makes full use of the processor performance, and therefore it could reach a high conversion speed, that is to finish in 1/10 to 1/2 broadcast time.

Using MEncoder

The basic syntax of MEncoder is: mencoder inputfile[file|URL|-] [-o outputfile] [options], where “inputfile” is name of the source video files with the full path (the file in hard disk or Webpage address), “-o” is the output file command, “outputFile” is the name of the output files with the full path; “options” is more complex, and some of the MEncoder options are listed in table 2 below.

Table 2 MEncoder common options

Options	Function Description
-of	Select the output video container format
-oac	Specify the coding audio encoder
-ovc	Specify the coding video encoder
-lameopts	Set the audio encoder
-lavcopts	Set the video encoder
-af	Configure the audio filter
-vf	Configure the video filter
-srate	Set the audio sample rate
-ofps	Output of the video frame

MEncoder video format technology

The MEncoder command can convert video format. Video format conversion command, as shown in Figure 3, converts the “input.avi” file to “output.flv” file. It uses mp3lame as the audio encoder encoding, and LAVC (Libavcodec) as the video encoder. Since Libavcodec contains many kinds of video coding, vcodec uses FLV coding. This command set the audio coding rate to 56kbps, the audio sample rate to 22050Hz, and the video coding rate to 500kbps.

```
mencoder input.avi -o output.flv -of lavf \
-oac mp3lame -lameopts abr:br=56 -srate 22050 -ovc lavc \
-lavcopts vcodec=flv:vbitrate=500:mbd=2:mv0:trell:v4mv:cbp:last_pred=3
```

Figure 3 MEncoder video format technology

The comparison of the two kinds of video format conversion technology

Comparison of function

The FFmpeg command line can add options, specify the output format, support the encoding and decoding of the input file, and intercept single frame image from the video, while MEncoder does not support these functions^[6]. In the area of video compression, the two are similar. The main difference is that MEncoder can edit all FFmpeg code supported by the libavcodec^[7], and its video format is more complete than FFmpeg. For example: MEncoder could convert files with extension named.Rm,.Rmvp,.rt which FFmpeg does not support^[8].

Comparison of Efficiency

In this experiment, FFmpeg and MEncoder conversion software are runned in the Ubuntu 12.04 operating system, and they convert the same HD video respectively. Then we test the difference of their converting rate from two angles of time used in converting video and the output video size. We use 100 HD video clips whose time length is more than 1h20mn, coding rate is 3535Kbps, and the coding format is MPEG-4. They are mainly the ski movement pictures and character narrative pictures, which can effectively detect the quality of video trans coding when compressing dynamic and static pictures. Both FFmpeg and MEncoder use FLV, which is mentioned in the command line in Table 3, for video compression. They use mp3 for audio compression, set the audio sample rate to 22050Hz, set the video bit rate to 3000Kbp, set the frame rate to 24fps, and the container is flv. Test results and analysis are as follows:

For example, the extracted source skiing video file size is 1.99GB (2137663152 bytes), and its size after trans-coding is shown in table 3. In order to maintain the HD video definition after transcoding, we set a high rate. So there isn't much change in the transcoded video size compared to the source file, but the MEncoder has a slightly higher compression ratio in the same option parameters. In aspect of video conversion time, FFmpeg uses less time, and it has a fast coding speed. In aspect of the picture effect, the output video clarity has a certain degree of decline due to the compression loss, and the

transcoded video quality of MEncoder is slightly higher compared to FFmpeg. 92% of the 100 test videos are in line with the law in this experiment. It is proved by some scholars that, blurred screen phenomenon often appears in FFmpeg when converting “.Wmv” and “.Asf” files, and its Mkv format conversion effect is poor^[8].

Table 3 Comparison of conversion efficiency between FFmpeg and MEncoder

Conversion Technology	Command Line Options	Conversion Time	Video Size
MEncoder	-of lavf -oac mp3lame -lameopts abr:br=56 -srate 22050 -ovc lavc -lavcopts vcodec=flv:vbit rate=3000 -ofps 24	1972.45s	1.72GB (1849468065 bytes)
FFmpeg	-f flv -vcodec flv -r 24 -b 3000000 -acodec mp3 -ar 22050 -ab 56	1870.21s	1.72GB (1854874191 bytes)

In summary, FFmpeg has the advantages of high encoding speed, more comprehensive functions and more flexible usage, and the MEncoder can convert a variety of videos with high quality, but the conversion speed is slower than FFmpeg. Therefore, in the video file format conversion, we could use the two kinds of technology at the same time, considering the video encoding information of the two encoders, the video format, and their encoding time. First of all, according to the video format set classification shown in table 4, the format like WMV, RM, RMVB, RT, which FFmpeg can't handle or not well, could be processed by the MEncoder. Secondly, to enhance the quality of video after conversion, video is classified according to their length, and the larger will (e.g, lesson) be handled by the FFmpeg in order to reduce the conversion time, while other smaller (such as, "micro class") will be processed by the MEncoder, ignoring the slight conversion difference.

Table 4 The Video Format Fet Cassification ForVideo Conversion ^[8]

Conversion Components	Convertible Video Format
FFmpeg	*.mov *.avi *.mpg *.mpeg *.mpe *.3gp
MEncoder	*rm *.rmvb *.rt *.wmv *.asf *.asx *.mkv

Module design for video format conversion

The format FLV has become one of the most popular streaming media formats widely used in video website, because of its minimal size, fast loading speed and its feature that it could be played by FlashPlayer in the Webpage. The video format conversion module in this server uses both the FFmpeg and MEncoder at the same time, and it will convert the formats of all video uploaded to the server to the FLV video format. The function design process of the whole video format conversion module is shown in figure 4.

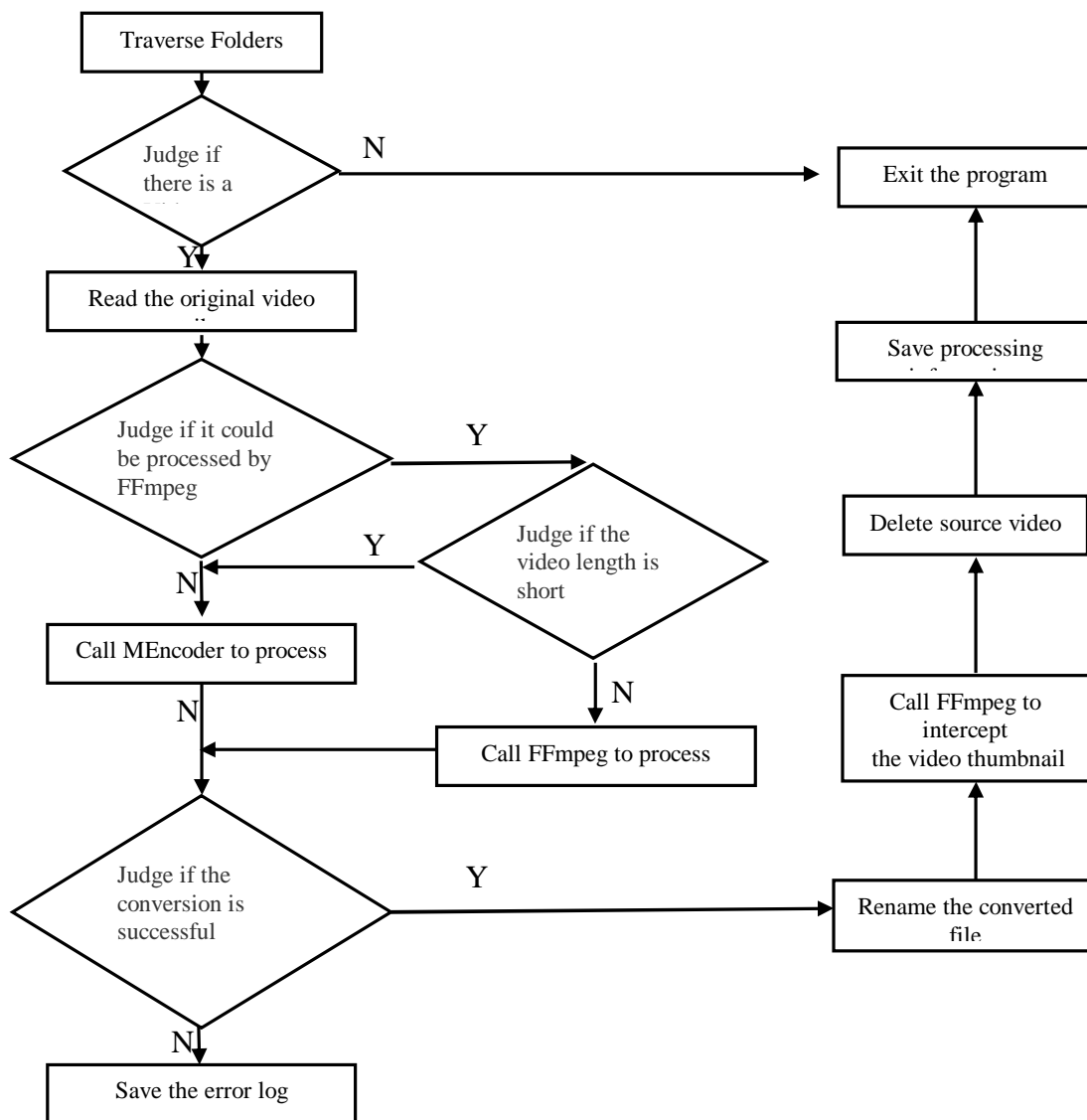


Figure 4 Video Format Conversion Module Process

Traverse video Folders. Judge if there is a video file. If there is a video, read the video attribute information using the FFmpeg command, including the length, format, resolution, frame rate, bit rate and other parameters, and judge whether its format is belong to the FFmpeg format set according to table 4. If it belongs to FFmpeg, it is judged whether the video length is shorter, and the shorter calls the MEncoder to convert the video according to the video attribute information, the longer calls FFmpeg to convert the video. if it doesn't belong to the FFmpeg, then call the MEncoder to convert the video according to the attribute information of video. When the video conversion is completed, judge whether the video conversion is successful. If the conversion is successful, then rename the file, call the FFmpeg to intercept video thumbnails, delete the source video file, and save processing information, etc. If failed, error messages are written to the error log, and then traverse video folder again. Exit the program if there isn't any video file after traversing the finished video folder.

Conclusion

There are a variety of video formats for the present EVOD system becomes more and more open. In this paper, through the test, the analysis and comparison of two kinds of the most popular video

compression and processing software FFmpeg and MEncoder, and combining their advantages and video coding information, this paper designs a server video format conversion module, providing unified video file format technical solutions for the EVOD system developers. As the rise of cloud computing and cloud terminal popularity, the VOD system is no longer satisfied with the positive conversion of different upload video formats to a format, and it tends to use the cloud server to get the best effect of video conversion according to the network link status and the terminal screen resolution calculation. This study will further study the EVOD system based on cloud computing, to provide the best results for the current terminal for the educators and the educated.

Acknowledgements

This work was financially supported by the Major Projects of Guangdong Province Science and Technology(No.[2012]99), Zhuhai Education Research Projects(2014D0401990065).

References

- [1] Wu Ai,Liu Xinsong,Fu Qingyun,and Liu Kejian. DPVoD:P2P Based Video-on-Demand Architecture. Journal of Computer Research & Development. 2008, 45(2):269-277.
- [2] SHEN Shi-Jun LI San-Li. P2P-Based Video-on-Demand Systems: A Survey. Chinese Journal of Computers, 2010(04)
- [3] Zhiyong Jin, Chuanle Sun, Zhiming Song. Construction and application of campus network based video on demand system in teaching. E-education reseach, 2009(02).
- [4] Ishfaq Ahmad , Xiaohui Wei , Yu Sun , Ya-qin Zhang, Video Transcoding: An Overview of Various Techniques and Research Issues, IEEE TRANSACTIONS ON MULTIMEDIA, MULTIMEDIA, 2005, vol. 7, p. 793--804.
- [5] Shaoyong Chen, Min Lin, Shifu Liang. Design and implementation of open source software-based video-on-demand system. China Education Info,2011(03):30-32.
- [6] Berton D. Advanced video coding on Linux[J]. Linux Journal, 2006 (150): 74-77.
- [7] Heikki Orsila, Jaco Geldenhuys, Anna Ruokonen, Imed Hammouda, Trust issues in open source software development, WUP '09 Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010, P. 9-12.
- [8] FFmpeg Documentation. <http://www.ffmpeg.org/documentation.html>. 2010.