

Parallel video transcoding using Hadoop MapReduce

Mingang Chen^{1, a}, Wenjie Chen^{1, b}, Zhenyu Liu^{1, c} and Lizhi Cai^{1, d}

¹ Shanghai Key Laboratory of Computer Software Testing and Evaluating, Shanghai Development

Center of Computer Software Technology, Shanghai, 201620, China

^acmg@ssc.stn.sh.cn, ^bcwj@ssc.stn.sh.cn, ^clzy@ssc.stn.sh.cn, ^dclz@ssc.stn.sh.cn

Keywords: video transcoding; Hadoop; MapReduce; FFmpeg; parallel transcoding.

Abstract: Video transcoding has become a key technology for video content distribution network service. In this paper, we propose a novel MapReduce-based parallel video transcoding method. In our method, video files are stored on a shared file system to reduce the overhead of disks I/O and networks in the Hadoop MapReduce. FFmpeg is used to compute the splitting point of the video and the actual video transcoding. Experimental results show that our method can significantly reduce the time of transcoding.

1. Introduction

With the rapid development of network video applications, how to support the access to video content of various terminal equipment (PCs, laptops, smartphones, etc.) has been an important issue concerned by the CDN(content distribution network) service providers. As we all know, video player terminals are heterogeneous, such as different CPU processor capabilities and screen resolution, so video format requirements on them are not the same, including the encoding format, media containers, video and audio bit rate, frame rate, resolution and so on[1]. Therefore, video transcoding has been studied and applied to adapt to various terminals. In addition, in the video CDN, edge nodes convert video bit rate and resolution taking into account the network bandwidth, delay, jitter, packet loss rate of the user terminal devices to ensure the user's watching experience.

Video transcoding, the video from one format to another format, including compression algorithm, resolution, bit rate, frame rate [2]. Video transcoding is not only the core technology of video CDN, but also widely used in IPTV, cloud storage, cloud download, etc. With the popularity of high-definition video and the rapid development of video on demand, live video service, video transcoding in the single machine cannot meet the needs of large-scale video transcoding. Distributed video transcoding can make full use of the advantages of parallel computing to reduce the transcoding time, and distributed system is scalable.

Currently, a popular distributed computing model-Hadoop MapReduce[3] has been widely used in the field of distributed video transcoding[4-6]. However, storing video files on Hadoop HDFS can incur the overhead of disks I/O and networks. Furthermore, MapReduce does not inherently support the splitting of the video file. So you have to modify the FileInputFormat class in Hadoop MapReduce or modify the transcoding software to support reading and writing files in HDFS.

In this paper, we propose a novel parallel video transcoding method, which uses original MapReduce and FFmpeg [7] to achieve efficient distributed transcoding. In our method, video files

are stored on a shared file system to reduce the overhead of disks I/O and networks in the MapReduce. Video splitting is no need to actually split a video, just computing time code of video's splitting point by FFmpeg and recording the time code.

2. Architecture of parallel video transcoding system

Parallel video transcoding system is composed of the application server, shared file system, and Hadoop cluster. The latter two are managed by cloud management software CloudStack. The system can be divided into three layers logically, as shown in Figure1. From bottom to top, they are computing and storage layer, application service layer and portal layer.

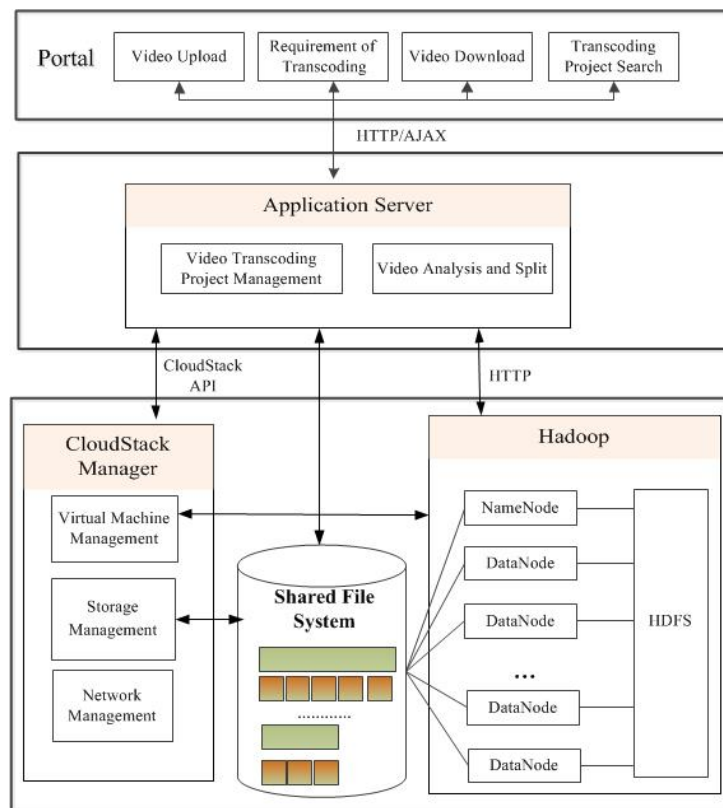


Fig 1. The framework overview of parallel video transcoding system

(1) Computing and storage layer: The Hadoop cluster takes the computing tasks of video transcoding. The storage system consists of two parts, in which the original videos and videos after transcoding are stored in the shared file system, while the video transcoding task file (configuration file) is stored in the HDFS of Hadoop cluster.

(2) Application service layer: The core function of application service layer is to start and schedule the jobs of video transcoding. Usually, the user sends a transcoding requests to the web application server, then application server submits the job of transcoding to Hadoop namenode according to the user's transcoding job requirement. Namenode assign the transcoding job to the multiple datanode, and datanode call FFmpeg to execute the parallel video transcoding.

(3) Portal layer: Portal provide users with a variety of business functions, such as the upload of the original video, the setting of transcoding requirement template, and the query of the transcoding job.

3. Implementation of parallel video transcoding system

Parallel video transcoding mainly consists of two phases, the split of the video clip and the distributed video transcoding using Hadoop MapReduce, as shown in Figure 2.

3.1 The split of video clip.

The purpose of video splitting is to segment whole video clip into multiple sub-clips. Hadoop MapReduce schedules multiple mappers to execute FFmpeg transcoding command for these sub-clips. In our implementation, there is no need to actually split the video. We just compute and record the time code of the split point. This kind of parallel video transcoding can reduce the transcoding time cost of the whole video clip. As for the number of sub-clips can be decided by the number of datanode in Hadoop cluster. Usually, the number of sub-clips is equal to the number of datanode. It is worth noting that these split point must be the key frame of the video, otherwise it will cause the lack or the repetition of video frame after the combination of video clips in Reducer.

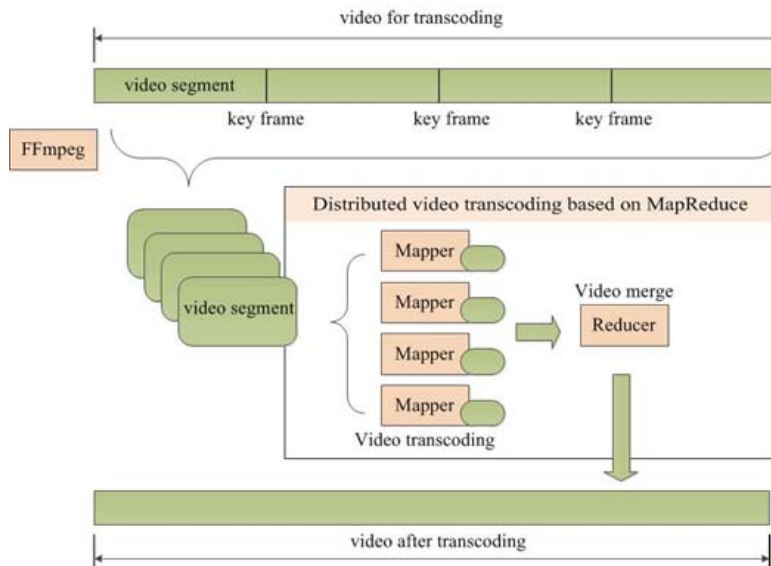


Fig 2. The framework overview of parallel video transcoding system

3.2 The distributed video transcoding using Hadoop MapReduce.

Distributed video transcoding based on Hadoop MapReduce includes two phases, namely, the Mapper phase and the Reducer phase. In the Mapper phase, the map function reads the transcoding configuration file from HDFS. Each row of the configuration file lists the path of each video sub-clip in the shared file system, the start and the end time code and transcoding parameters. The core of map function is to call a custom transcoding function to achieve video transcoding. It is worth noting that in the MapReduce job settings the parameter *MapReduce.input.lineinputformat.linespermap* should be set to 1, which makes each mapper transcode a video sub-clip. In the Reduce phase, the reduce function merges multiple video sub-clips into one video according to the number of sub-clip.

4. Experiment and discussion

In this paper, we compare the time efficiency between single machine video transcoding and

parallel transcoding in Hadoop. The experimental environment is a Hadoop cluster with four nodes, one of the nodes is namenode, the other three are datanodes, and the hardware configuration of the four nodes are all two 1GHz CPU and 16GB memory. The experiment data are three videos with different file size, as shown in Table 1. The video transcoding requirement information is shown in Table 2 and the experimental results are shown in Table 3.

Table 1. Video information for performance evaluation.

	Video1	Video2	Video3
Container	MP4	MP4	MKV
File size	8316KB	93800KB	5.46GB
Video encoding	H264	H264	H264
Audio encoding	AAC	AAC	DTS

Table 2. Requirement parameters for video transcoding.

Target container	avi
Video encoding	H264
Video encoding	1024Kbps
Audio encoding	mp3
Audio bit rate	128Kbps

Table 3. Time performance comparison between single machine video transcoding and parallel transcoding in Hadoop.

	single machine video transcoding (second)	parallel video transcoding in Hadoop MapReduce (second)
Video1	43	36
Video2	370	166
Video3	6273	2507

It is observed from Table 3 that the efficiency of parallel video transcoding is not particularly noticeable when the file size of the video is small, such as video1 (8316KB). This is mainly due to the time cost of the scheduling of Hadoop MapReduce and the merging of the video. However, when the file size of the video is large, such as video3 (5.46GB), efficiency improvement is obvious by the parallel video transcoding. The time cost of the parallel video transcoding is only 40% of the single machine video transcoding. If we configure more high-performance transcoding nodes in the cloud, the efficiency of video transcoding will be further enhanced.

5. Conclusions

With the high-speed development of Internet video services, it is an urgent problem for CDN providers to solve the technology of fast video transcoding. This paper proposes a novel parallel video transcoding method based on original Hadoop MapReduce and FFmpeg to achieve efficient distributed transcoding. The key point of the method is the splitting of video and the parallel transcoding, which significantly improve the speed of transcoding. In further studies, we are planning to study job scheduling of the video transcoding to improve the overall efficiency of the transcoding system.

Acknowledgements

This work is funded by Science and Technology Commission of Shanghai Municipality program (15511107003 and 16511101202).

References

- [1] Z. H. Li, H. Yan, G. Liu, et al, Cloud transcoder: Bridging the format and resolution gap between internet videos and mobile devices, Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video. ACM, 2012: 33-38.
- [2] I. Ahmad, X. H. Wei, Y. Sun, et al, Video transcoding: an overview of various techniques and research issues, IEEE Transactions on multimedia, 2005, 7(5): 793-804.
- [3] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Communications of the ACM, 2008, 51(1): 107-113.
- [4] M. Kim, Y. Cui, S. Han, et al, Towards efficient design and implementation of a Hadoop-based distributed video transcoding system in cloud computing environment, International Journal of Multimedia and Ubiquitous Engineering, 2013, 8(2): 213-224.
- [5] C. Ryu, D. Lee, M. Jang, et al, Extensible video processing framework in apache hadoop, 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, 2013: 305-310.
- [6] A. Garcia, H. Kalva, B. Furht, A study of transcoding on cloud environments for video content delivery, Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing, 2010: 13-18.
- [7] FFmpeg website, <https://ffmpeg.org/>.