The Design and Implementation of Campus Network Streaming Media Live Video On-Demand System Based on Nginx and FFmpeg

Bo She[1], Qiang Wang[1], Xiaoge Zhong[2*], Zhe Zhang[1], Zunying Qin[1] and Guodong Li[1]

[1] The Network Information Center, Xi'an Jiaotong University, Shaanxi, Xi'an, 710049, China
[2] State Key Laboratory of Astronautic Dynamics, Xi'an Satellite Control Center, Xi'an, China
Email: shebo@xjtu.edu.cn

Abstract. For the current demand for live broadcast and live video in colleges and universities, relying on the advantages of campus network, this paper designed and implemented a video live broadcast and video on-demand system suitable for colleges and universities. The system used the streaming media server built by the open source Nginx to capture high-definition video streams output by cameras, etc. through the RTMP protocol, and can provide video live broadcast and video on-demand services using multiple carrier lines. Video live broadcast used the open source Flash player JW Player, stored video in high-performance HDFS distributed storage at a certain time interval and format by a storage algorithm, and supported on-demand playback of MP4 format video stored in HDFS. The paper tests the various functional modules of the system and the overall performance of the system to verify the effectiveness of the system. The system is currently deployed in the campus network of Xi'an Jiaotong University, providing normal recording and broadcasting of video of 293 intelligent classrooms in Xi'an Jiaotong University, video on-demand for after-school courses, video live broadcast and video on-demand for various large-scale events held on campus.

1. Introduction

As the pace of university information construction continues to accelerate, the network infrastructure of university is continuously improved and the scale is continuously expanded and upgraded, the management and education of university are becoming more informationized. The Internet has made colleges more information-oriented, diversified, and modernized. Among them, live video on-demand is the media form that can best reflects the real-time transmission of information.

The rapid development of information technology has provided a variety of technical means for university in education, management, and publicity. The application of live video on-demand technology in universities is mainly reflected in three aspects.

(1) In terms of teaching, it can break the limitation of the number of students, expanding the traditional teaching classroom, and students can remotely self-learn, such as famous teachers' lectures, master lectures and courses review video after class.

(2) In terms of teaching management, it provides strong technical support for teaching supervision. For example, the supervisor can check teaching remotely in real-time, and teachers can check the quality of teaching after class, etc.

(3) In terms of school publicity, it provides intuitive and powerful information promotion channels, such as live broadcasts of various large-scale activities (celebrations, graduation ceremonies, etc.) on the web, school publicity on-demand video applications, etc.

For the current demand for live broadcast and live video in colleges and universities, relying on the advantages of the campus network, this paper has designed and developed a set of streaming media video on-demand system suitable for universities which can flexibly meet the demands of colleges in education, teaching, teaching supervision and activities hosting by real-time live broadcast. The system uses load balancing, video codec, distributed storage, multiple operators' lines and other technologies to provide users with a 720p live on-demand viewing experience.

2. The Design and Implements of the System

At present, most campus network construction has entered the era of high-speed networks for end users. Although the campus network export bandwidth is fully loaded, the internal network traffic is small and has greatly redundant. This system is designed using the advantages of existing campus network internal link bandwidth redundancy. The overall framework module of the system is shown in figure 1 below. The whole system is divided into three modules, which are streaming acquisition and storage modules, streaming on-demand module and streaming live module.

The streaming acquisition and storage module captures video streams by cameras automatically. It uses the RTSP protocol to capture video streams for the cameras in the classroom. For SDI/HDMI HD video streams output by cameras and other equipment, it uses HD video encoding and decoding to convert HD video stream to standard H.264 [1-2] high-definition video stream with a certain bit rate and then collected by a streaming media receiving server. The collected high-definition video stream is stored into a specified high-performance HDFS distributed storage through a storage algorithm.

The stream live module will decode and shunted received video stream for multiple operator lines. The live broadcast uses the open source Flash video player JW Player, which supports users to watch videos using mobile phones and computers at the same time. The system uses a load balancing server to divides users into different streaming media servers for live video streaming intelligently according to server load. In order to decrease the latency of live video streaming, the RTMP protocol is used for live video streaming [3-4]. The stream on-demand module is used for providing on-demand video stored in HDFS distributed storage. The server used in the streaming on-demand module is Nginx which support on-demand MP4 format videos stored in HDFS.

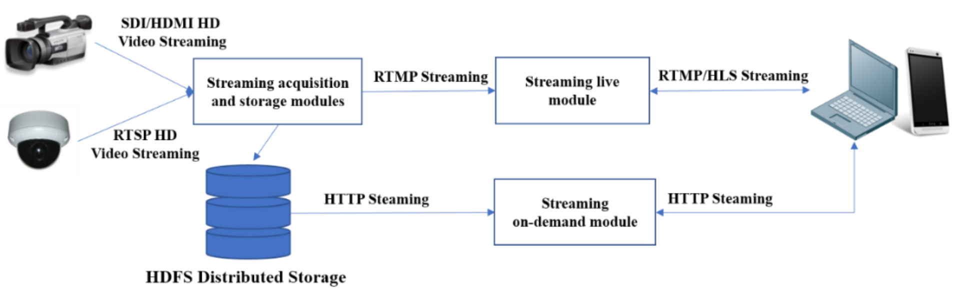

Figure 1. System framework module division diagram.

2.1. The Streaming Acquisition and Storage Module

The stream acquisition and storage module of this paper mainly collects and stores the video and audio signals captured by various cameras or the high-definition video signals output by professional directors. Generally, cameras and professional directors use HDMI interfaces, SDI interfaces, the RTSP or RTMP protocol for signal output, so the system needs to use professional signal acquisition and coding hardware equipment for the collection of video signals for various events held in the school. The classroom camera and encoder video stream collection use the standard RSTP protocol. The system needs to store videos from 293 smart classroom cameras. Each classroom has three 720P video signal outputs. About 900 video signals need to be collected and stored at the same time. Therefore, a high-performance, large-capacity and scalable distributed storage system is needed to store massive video resources.

2.1.1. Multi-threaded Parallel Acquisition Storage. This paper needs to collect and store videos for teachers' courses and various activities held in the school. At present, Xi'an Jiaotong University has built 293 intelligent classrooms. Each classroom is equipped with 2 cameras and 1 computer VGA encoder. Each unit's video stream output is 2Mbps, and the system needs to meet the collection and storage of $2\text{Mbps} * 3 * 293 = 1.72\text{Gbps}$ video stream. With the fact of such a high-speed video stream, in order to ensure normal acquisition and storage, the system uses multi-threaded parallel acquisition technology.

A single classroom can generate $(2\text{Mbps} * 3 * 3600) \div (8 * 1024) = 2.6\text{GB}$ of data each hour. Based on 5 lessons per day for 10 hours, the amount of data generated by 293 classrooms per day is$2.6\text{GB} * 10 * 293 = 7.44\text{TB}$. The university requires classroom videos to be stored for half a year. Such a large amount of storage places high requirements on the system's storage performance and capacity. Traditional NAS storage cannot meet the requirements regardless of performance and capacity. This paper uses Hadoop-based distributed file storage. The HDFS's high fault tolerance can meet the requirements for data storage security, high-throughput data access meets the system's high storage performance write requirements, and it can be deployed using inexpensive ordinary servers, which greatly reduces the system's storage cost.

The Hadoop-based distributed file storage system HDFS is based on Hortonworks' open source data platform [5-11]. The Hadoop platform is deployed with HDFS, YARN, Mapreduce2, HBase, Zookeeper, Spark2, Ambari and other components. The system uses 15 servers. Each of them is installed with two 500GB SAS disks as raid1 as system disks and 16 6TB SATA disks as data disks. The total storage capacity of the system is 1.2PB.

The streaming media acquisition server uses the open source FFmepg software to receive video streams. FFmpeg is an open source cross-platform software that can be used to record, convert digital audio and video, and convert them into streams. FFmpeg supports multi-threaded and multi-process. The system architecture for video streaming acquisition and storage module in a single classroom is shown in figure 2. The video acquisition of two cameras and one VGA capture box in a single classroom uses the standard RTSP protocol. The capture thread is collected through the classroom-specific IoT three-way RTSP video stream. In order to avoid the low performance of HDFS distributed storage caused by a large number of small data writes, the capture thread writes the video data to the circular loop buffer. When the data packets in the circular loop buffer accumulate to a certain amount, the storage thread writes the data packets in the cache to the HDFS distributed storage in a certain format at a time.

In order to improve the performance of the system's collection and storage performance, the system uses multiple servers to store the video streams of 293 classrooms, and each server only collects and stores the video streams of some fixed classrooms.

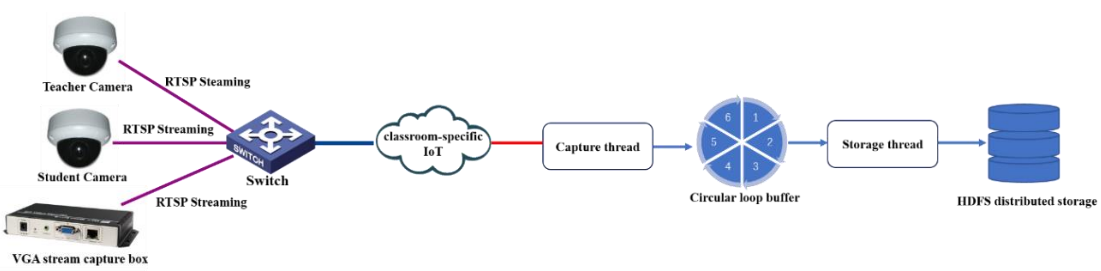

Figure 2. Single classroom video stream acquisition and storage implementation architecture diagram.

2.1.2. Circular Loop Buffer. We introduced a circular loop buffer [12] in the streaming acquisition and storage module. The capture thread and storage thread of each classroom share the circular loop buffer.

The capture thread writes data to the circular loop buffer, and the storage thread reads from the circular loop buffer and then write them into HDFS storage [13-14], The storage thread needs to wait until the data in the circular loop buffer reaches a certain size before writing to the storage, so as to avoid frequent writes of small data volumes leading to the low performance and greatly improves the system performance.

The form of circular loop buffer in memory is a continuous area of size N * M where N is the number of memory blocks (BUFF_NUM), and M is the size of the memory block (BUFF_SIZE). Since HDFS is written as a block by default, the system's storage block size is 128MB, the default block size of the circular loop buffer is 256MB, and N is set to 6. When flag of each circular is empty, it can be written data. When the capture thread writes a data packet, it writes in sequence according to the number of the memory block 0 => 1 => 2 => 3 => 4 => 5 => 0, like a circular loop. All memory blocks appear as a ring-shaped memory space connected end to end, as shown in figure 3 below. Each time a memory block is filled, set a flag for this memory block as full. It means that the memory block is not writeable and can only be read. The storage thread also sequentially reads the data packets according to the number of the memory block 0 => 1 => 2 => 3 => 4 => 5 => 0, which is also cyclic.

When the memory block pointed by the storage thread is empty, it indicates that there is no data can be read. The storage thread will hang and wait for the pointed memory block to be full. When the memory block pointed is full, the storage thread reads the data packets in it and writes them to the stream file. After writing all the contents of the memory block to the stream file, the memory block is marked as empty, and the storage thread loops point to the next memory block.

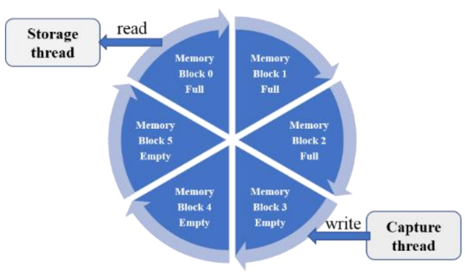

Figure 3. Circular loop buffer.

2.2. The Video Streaming Live Module

The video streaming live module is used to broadcast various activities held inside and outside the university and classroom lectures. The system architecture of the video streaming live module is shown as figure 4 below. The video streams for various activities are generally SDI/HDMI HD streams. The teaching video stream is RTSP HD stream. The SDI/HDMI HD stream is encoded using an HD video encoder and converted to an RTMP format stream. The RTSP/RTMP HD stream is captured by FFmpeg, and the video stream which encoded as an adjustable RTMP steam with different bit rates is pushed to multiple streaming media servers. Each streaming media server uses a different export line. The university currently uses 4 servers for live video streaming. The line uses CERNET lines, Telecom lines, Mobile lines and Unicom lines.

The streaming server uses the RTMP module of the open source software Nginx. Nginx [15] is a lightweight web server/reverse proxy server. It is famous for less memory and strong concurrency. It not only can be used as a proxy server, but also can be used as a load balancing server [16-17]. Nginx-rtmp-module is an extension module of Nginx which is able to turn Nginx into a streaming media server that can support the RTMP protocol. It can support RTMP and HLS (Live Http Stream). Users can choose the appropriate streaming media server for video-on-demand viewing according to the carrier network currently used. The system supports mobile phones and computers to watch at the same time.

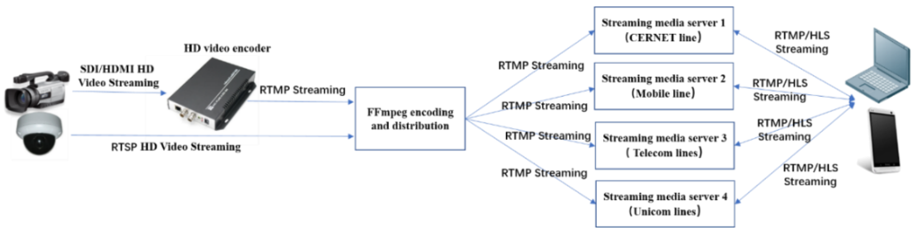

Figure 4. Video live broadcast system architecture diagram.

2.3. The Video Streaming On-Demand Module

The video-on-demand module provides on-demand services for daily lectures in the classroom and for videos stored during various activities held inside and outside the school. The system architecture implemented by the video-on-demand module is shown in figure 5. The system builds Nginx load balancing server at the front to shunt user's on-demand requests to different streaming media servers. The system builds 4 streaming media servers and uses different operator exit lines, including CERNET, China Telecom, China Mobile, China Unicom. The Nginx load balancing server shunt video on-demand requests to the streaming media server corresponding to the operator's export lines according to the user request's IP address. The streaming media server is built using Nginx's Http_mp4_module module. It supports MP4 format video on-demand services. Because all video files are stored in HDFS distributed storage, loading the video file by the streaming media server is a major problem. In order to facilitate the loading of the video file by the streaming media server, the system uses fuse to mount HDFS file system. Fuse is a module for mounting certain network space to the local file system in Linux. And it also supports mounting of the HDFS file system. The system uses fuse to mount HDFS storage on each streaming serve. For a streaming media server, HDFS storage is equal to local storage after mounted, which solves the problem of video loading in HDFS distributed file system of Hadoop.

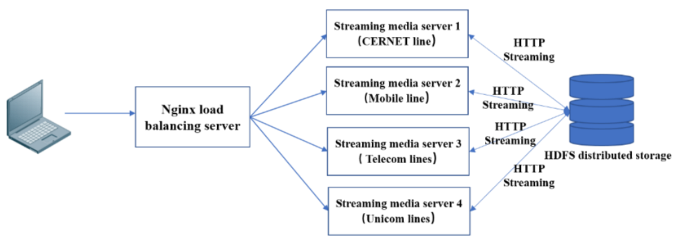

Figure 5. Video on-demand system architecture diagram.

3. The System Testing and Implementation

This paper tests the system's various modules and the overall performance of the system in detail. The system on-demand test mainly stresses the streaming server. It uses the Apache Bench ab tool to test the video access failure rate under large concurrent requests. The CPU usage, memory usage, and HDFS read capability are shown in table 1 below.

Table 1. Ab tool multiple test video on-demand results table.

Test Case	Requests sent in a single time	Total number of requests sent	Failed request	Streaming server CPU usage	Streaming server Memory usage	*HDFS read bandwidth*
Case1	10	10	0	5.1%	10%	10Mbps
Case2	100	100	2	7.3%	15%	100Mbps
Case3	1000	1000	5	10.2%	25%	1000Mbps

From the results in table 1 we can see that the CPU usage and memory usage of the streaming media server have a small impact on the system during the system on-demand process. The HDFS data read capability can meet the needs of multiple concurrent video on demand. The current system can meet at least 3,000 videos on demand concurrency.

The system live tests the effect of the video live function and the pressure of the streaming media server. Because the live system uses the open source player JW Player, it is difficult to test with tools. Therefore, the test is performed in a high concurrent user environment that provides video live services. The results of multiple tests show that the pressure of live video streaming mainly depends on the export bandwidth. The CPU and memory usage of the streaming media server for multiple concurrent users are small.

The application deployment structure of the system in the campus network of Xi'an Jiaotong University is shown in figure 6 below. The system has been successfully deployed and operated for more than 2 years. The system is mainly used for the normal recording and broadcast of class videos in all intelligent classrooms, including live classroom lectures, on-demand courses for students after class, and at the live service for large-scale activities held in the university. The current storage of HDFS is 1.2PB, which can meet the storage needs of all smart classroom lectures for three months. The system has been used for providing 720p live video on-demand services during graduation ceremonies, school celebrations, party congresses, teacher congresses, and various academic lectures which has been well received by teachers and students.

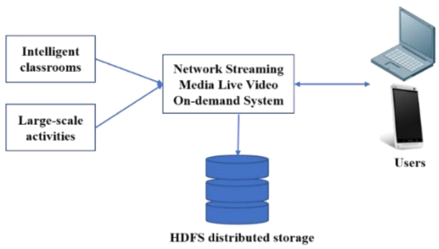

Figure 6. System deployment structure diagram.

4. Conclusion

This paper aims at the current needs of universities in education, teaching, publicity, teaching management and other aspects of live streaming video on demand. Based on the advantages of the campus network, a set of live streaming video on demand system is designed for our university. The system is currently deployed in the campus network of Xi'an Jiaotong University and is equipped with multiple operator line exports. The system provides normal recording and broadcast of class videos for 293 smart classrooms at Xi'an Jiaotong University, and videos on-demand for students' after-school courses. The system has provided 720P live on-demand services for various large-scale events such as the graduation ceremony of Xi'an Jiaotong University, the celebration of Xi'an Jiaotong University, and the Party Congress of Xi'an Jiaotong University. Each live has more than 5,000 people to watch. the system provides new media means for education and teaching, teaching management, and school publicity.

Acknowledgments

This work was supported by CERNET Innovation Project (NGII20190502).

References

[1] Nurrohman A and Abdurohman M 2018 High performance streaming based on H264 and real time messaging protocol (RTMP) *2018 6th International Conference on Information and Communication Technology (ICoICT)* (Bandung) pp 174-177.

[2] Santos H, Rosário D, Cerqueira E, Camargo J, Schimuneck M, Nobre J and Both C 2016 A comparative analysis of H.264 and H.265 with different bitrates for on demand video

streaming *Proceedings of the 9th Latin America Networking Conference* (*LANC'16*) (Association for Computing Machinery) (New York, USA) pp 53-58.

[3] Aloman A, Ispas A I, Ciotirnae P, et al. 2015 Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in mobile networks *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)* (Munich) pp 144-151.

[4] Li X and Jiang X 2012 Design and implementation of streaming media processing software based on RTMP *2012 5th International Congress on Image and Signal Processing* (Chongqing) pp 192-196.

[5] Lu K, Dai D and Sun M 2013 "HDFS+: Concurrent Writes Improvements for HDFS *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing* (Delft) pp 182-183.

[6] Changtong L 2016 An improved HDFS for small file *2016 18th International Conference on Advanced Communication Technology (ICACT)* (Pyeongchang) pp 474-477.

[7] Zhang J, Sun H and Yuan S 2019 HDFS copy storage improvement strategy *Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing (ICCBDC 2019)* (Association for Computing Machinery) (New York, USA) pp 71-77.

[8] Bhathal G S and Dhiman A S 2018 Big data solution: Improvised distributions framework of Hadoop *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (Madurai, India) pp 35-38.

[9] Divyashikha S, Shalini S and Huzur S 2017 Optimized MapFile based Storage of Small files in Hadoop *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'17)* (IEEE Press) pp 906-912.

[10] Sanjeev P, Ankur S and Dharmender S K 2014 Implementation of distributed searching and sorting using Hadoop MapReduce *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (ICTCS'14)* (Association for Computing Machinery) (New York, USA) Article 77 pp 1-5.

[11] Bansal N, Upadhyay D and Mittal U 2014 Concurrency control techniques in HDFS *2014 5th International Conference - Confluence the Next Generation Information Technology Summit (Confluence)* (Noida) pp 87-90.

[12] Chen D, H Li and Y Yuan 2007 Based on circular loop cache and precise timing multi-threaded data acquisition *Microcomputer Information* (34) 113-115.

[13] He J, Chen H and Hu F 2015 ERP: An enhanced read policy for HDFS to improve read performance for files under construction *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)* (Nanjing) pp 353-357.

[14] Patel Nhha M, Petal Narenddra M, Hasan M I, et al. 2014 Improving HDFS write performance using efficient replica placement *2014 5th International Conference - Confluence the Next Generation Information Technology Summit (Confluence)* (Noida) pp 36-39.

[15] Wen Z, Li G and Yang G 2018 Research and realization of Nginx-based dynamic feedback load balancing algorithm *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Chongqing) pp 2541-2546.

[16] Chi X, Liu B, Niu Q, Wu Q, et al. 2012 Web load balance and cache optimization design based nginx under high-concurrency environment *2012 Third International Conference on Digital Manufacturing & Automation* (GuiLin) pp 1029-1032.

[17] Zhang H, Lu N and Li J 2019 Load balancing based on node potential energy for distributed environments with limited migration capacity *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* pp 1479-1482.