Design and Implementation of Living Streaming System based on Multi-service Nodes Collaboration

Huang Tianchi
School of Computer Science and Technology
Guizhou University
Guiyang, Guizhou
zpli1@gzu.edu.cn, me@mythkast.net

Li Zeping,Zhang Jun
School of Computer Science and Technology
Guizhou University
Guiyang, Guizhou

***Abstract*—This paper presents a design and implementation method of living streaming system based on multi-service nodes collaboration. The system uses the entropy weighting method to dynamically determine the weight factor of each service node and uses the coordinator node to collect the load information of service nodes and perform global optimal distribution of loads. The collection terminal uses RTMP (Real Time Messaging Protocol) as the push protocol to send videos to service nodes, which then transcode the received videos into HLS (HTTP live streaming) and forward them to the releasing nodes in real time. Users can watch live videos on the releasing nodes. This service model is particularly suitable for watching live videos on mobile terminals. Herein, the design and implementation of the proposed system is described and its feasibility and effectiveness are verified in actual network environments.**

***Keywords*—collaboration,HLS,RTMP,entropy weighting.**

I. Introduction

With the rapid development of mobile communication technology, there is a growing need for receipt of Internet information and services whenever and wherever possible on mobile equipment. In particular, with the development of mobile 4G technologies, etc., live audio and video broadcast has become one of the mainstream services offered by mobile operators. However, in the face of the growing service needs and video data, how can service providers provide users with high-quality live audio and video broadcast services efficiently? This issue is turning into a research hotspot.

At present, the mainstream commercially-available live streaming solutions are based on RTMP [1][2]. But most mobile terminals are not available for RTMP [3][4], while they support live broadcast with the aid of HLS-based browsers [5][6][7], so RTMP needs to be transcoded [8] into HLS [9] on mobile terminals in real time. See **Fig.1** for RTMP-HLS server architecture [10].The shortcomings of this architecture:

- Limited service capacity and poor scalability in single-node service mode.[11][12]
- Poor reliability in single-node service mode.[11][13]
- High overload [14] probability in single-node service mode.

This paper put forward a design and implementation method of living streaming system based on multi-service nodes collaboration. Entropy weight method [15] is adopted in the system to dynamically determine the weight factor of each service node and collect load information from the service nodes through coordinator nodes for global optimum allocation. Users first make a request for a tile address to the coordinator node concerned. Then the coordinator node identifies the current highest-performance service nodes by entropy weight method based on the current state of the service nodes collected and then returns; users make an active connection to the service nodes offered by the coordinator; finally the system achieves a splitting effect.

In this paper we designed and implemented this system. According to the simulation experiment, the service performance of this system under multiple service nodes is much better than under a single node or the traditional multiple nodes. With higher load stability, each node in the system can provide services in accordance with its own practical service capability. In addition, the feasibility and validity of this system have been validated in a real network environment.

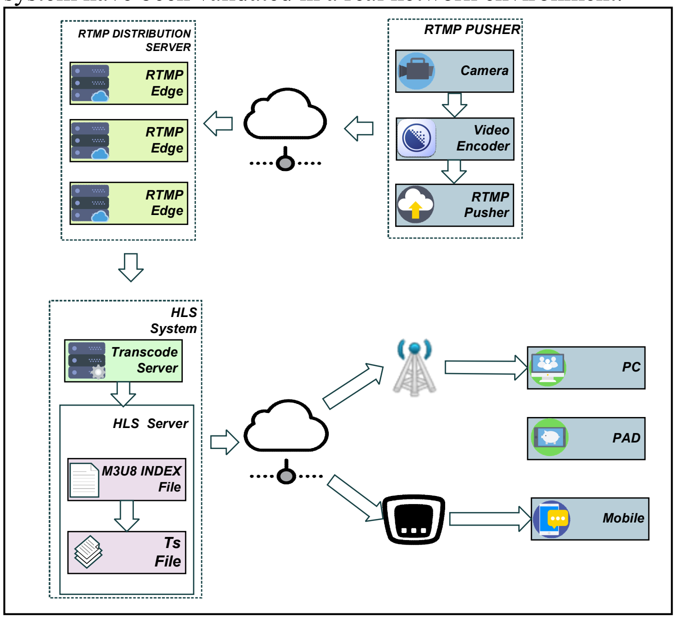

Fig. 1. Classic RTMP-HLS Server Architecture

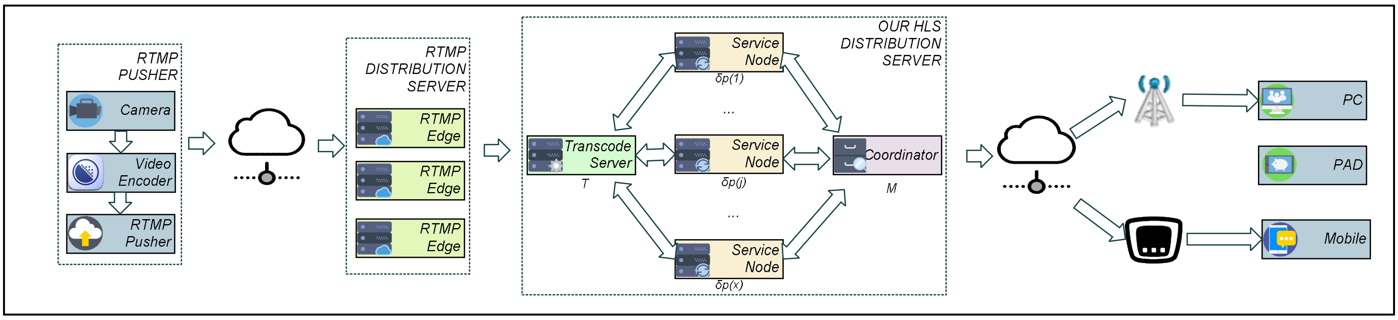

Fig. 2. Our Multi-Service Nodes Architecture

II. Design and Implementation of Program

A. Overall structure

Based on the defects of single-node service architecture, a multi-heterogeneous server load scheme is proposed. As shown in **Fig.2**, we suppose a service node group

$$P=\{<\delta p(1), \cdots, \delta p(x)>|x \in N\}.$$

After finishing video slice, transcoder T stores the video slice $s(i)$ into node $\delta p(1)$ and distributes it to the other nodes $\delta p(j)$. User u sends the video index file request to the coordinator first; the coordinator returns m3u8 index file according to the load of each node. After receiving m3u8 index file, user u connects to the specified node and downloads the required *ts* packet and play. Once the download is complete, a new index file from the coordinator will be requested again.

B. Entropy Weight Method

On multiple attribute decision problem, how to determine the index weight reasonability and objectively is extremely essential. Weight value reflects the importance of different indicators, and also have a direct impact on the results of comprehensive evaluation. At present, according to the source of the calculated data, the main methods of determining the weight can be divided into three categories: subjective weighting method [15][16], objective weighting method [17][18], and combination weighting method [19].

The system coordinator on the node group load evaluation indicators mainly consider the CPU usage r_c, memory usage r_m, bandwidth usage r_b and current traffic r_d[18]. The process of calculating the weight of the index by entropy method [15] is as follows:

The entropy of index j is

$$e_j = -k\sum_{i=1}^{n} y_{ij} \ln y_{ij} \qquad (1)$$

Where k is Boltzmann's constant ($k>0$ and $k=1/\ln(n)$, n is the number of evaluation objects), and the evaluation value corresponding to the index j of the object i after unification and normalization.

According to Eq.(1), the weight of indicator j can be defined as

$$\omega_j = \overline{e_j} \Big/ \sum_{j=1}^{m} \overline{e_j} \qquad (2)$$

Where m, is the number of selected evaluation indicators, and $\omega_j \in [0,1], \sum_{j=1}^{m} \omega_j = 1$. Differentiation coefficient, that is, when the greater the more important indicators, and ω_j can be calculated from Eq.(2). The server can finally get the evaluation value of the load:

$$L_i = \sum_{j=1}^{m} \omega_j r_{ij} \qquad (3)$$

In order to verify the performance of the system load balancer, is introduced to reflect the variance of the server load. Define the load balancing degree [19] as follows:

$$L_v = \frac{1}{n}\sum_{i=1}^{n} (L_i - L_{avg})^2$$

According to the real server load data, this paper takes the weight vector $\omega[r_c,r_m,r_b,r_d] = [0.196,0.088,0.450,0.266]$.

C. Function of Coordinator

```
Input:video_id,stream_index
for i=0 to 2 do
    for each rc,rm,rd,rb in Pi do
        Calculate Lpi based on Eq.(3)
    end for
    pi=min(Lpi)
    ts_url=pi..video_id..stream_index
    Add url into m3u8list
    Add stream_index
end for
return m3u8list
```

Fig. 3. Select Node Algorithm in Corrdinator

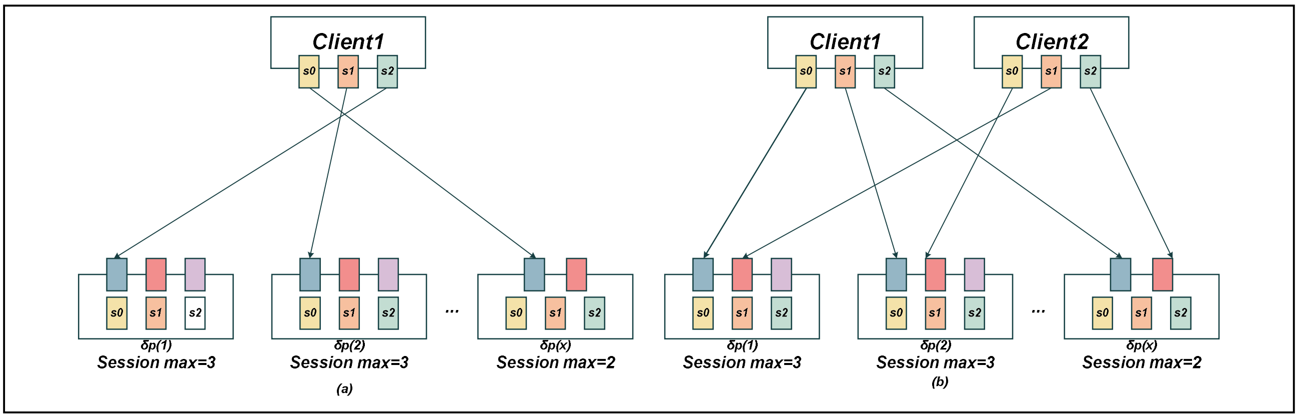

Fig. 4. Clients Connect Service Node via m3u8 File

Service node sends a heartbeat packet to the coordinator per unit time, and the packet includes current CPU usage, current memory usage, current traffic, and current bandwidth usage of the service node. As shown in **Fig.3** , according to the entropy weight method, the coordinator will dynamically select the optimal distribution node *δp(t)* via heartbeat packets provided by each server. Packet address will be written to m3u8 file in node *δp(t)*. According to the HLS specification, a common m3u8 file will have at least three recent slice time TS packet addresses[6].

D. Communication Between Service Node and User

On receiving the index file sent by the coordinator, the user accesses the file in specified distribution node according to the order of the detail of the file. As shown in **Fig.4** , with the increasing number of users, the service node will provide services according to the rules predetermined by the coordinator.

E. Protocol Stack Between Service Node and Corrdinator

Two main types of protocol stack have been established between the coordinator and service node, and both of them are based on HTTP protocol.

- After transcoder *T* finishes to slice video and generates to ts file, the coordinator will send a command means "ready to download" to each node. Each node will get new ts file from transcoder *T*. The protocol stack is shown in **Fig.5.**

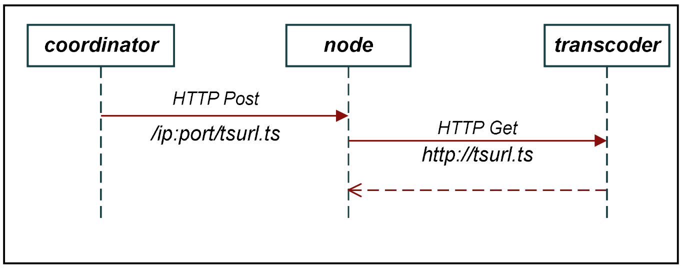

Fig. 5. Video Stream Request Protocol Stack

- Service node sends collected informations to the coordinator per unit time. The coordinator forwards the information to Redis [20] and waits for Redis asynchronous reply. Success information will be replied to the node server. The protocol stack is shown in **Fig.6.**

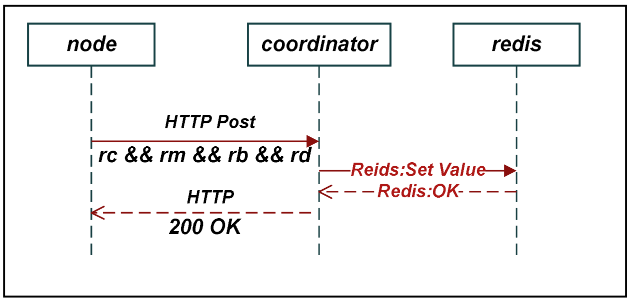

Fig. 6. Heartbeat Protocol Stack

III. EXPERIMENT AND SIMULATION

A. Setup Simulator

In order to verify the accuracy of the system, we develop a coordinator simulator based on multi-node coordination. The simulator is used to test the system. As shown in **Table I**, we first use simulator to simulate four servers named A, B, C,and D, then quantify the specific configuration of each server.

TABLE I. SERVICE NODE LIST

Node ID	Node Parameters			
	CPU	*Memory(M)*	*Bandwidth(M)*	*Max User*
A	100	5000	100	500
B	80	4000	100	300
C	50	4000	80	200
D	40	2000	80	100

We also give a change parameter when a user adds into the system, and it is shown in **Table II**

TABLE II. PEER PARAMETERS

Peer Parameters			
CPU	***Memory(M)***	***Bandwidth(M)***	***Max User***
0.2	1	0.2	1

B. Run Simulator

Simulator joins a user every unit time which is called *t*, and total time is 1100*t*. Each of the unit time simulators will be based on the various server parameters into the load formula and the user added to the specified service node, and calculate the variance of the four servers and recorded. The simulator tests the polling allocation algorithm, the random allocation algorithm and the load algorithm proposed in this paper. The simulator also records the number of users per service node per unit of time.

C. Results

The simulation results are shown in **Fig.7** and **Fig.8**. In **Fig.7** , x-axis means the unit time of the simulator's total simulation, and the ordinate is the variance size calculated by each of the four servers at each unit time. The results show that the select node algorithm proposed in this paper can achieve the best performance compared to the polling and random allocation. When a service node is close to be full-load, our algorithm proposed in this paper close to the ordinary round robin algorithm. In this case, a node server can be added to the system.

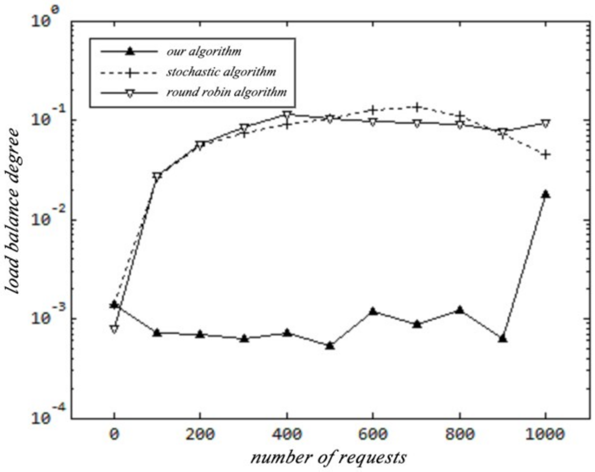

Fig. 7. Comparing With Other Algorithms

As shown in **Fig.8**, x-axis means the unit of time, and vertical axis means the server in the number of users per unit of time the size. The result shows that the load algorithm proposed in this paper fits the actual performance, for example, as the optimal performance of the server, node A withstand more user requests.

Our system is based on Linux operating system. To test out system, we use Microsoft Edge to connect out system via HLS. **Fig.9** shows that the screenshot of playing HLS in four Microsoft Edge browsers.

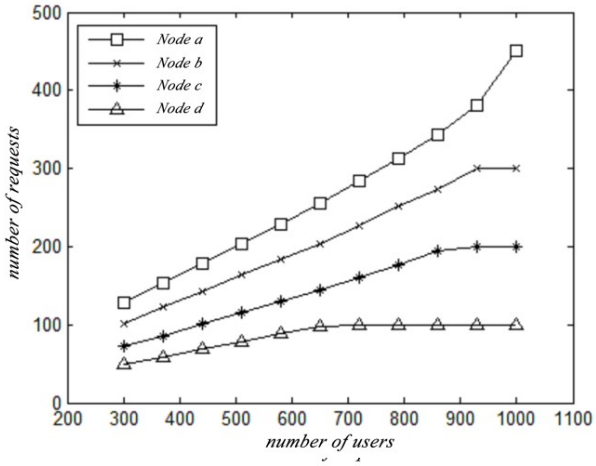

Fig. 8. Service Node State

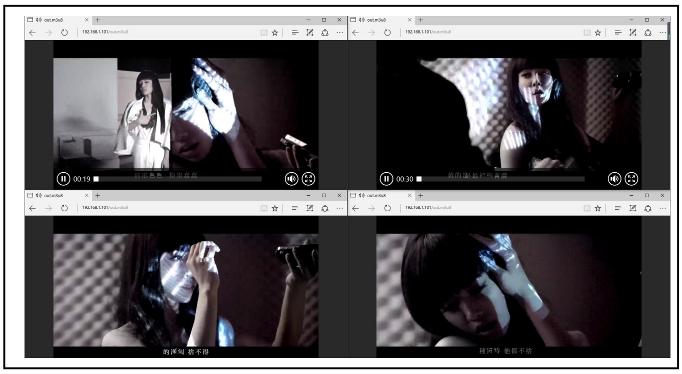

Fig. 9. Playig HLS in Four Microsoft Edge Browsers

IV. CONCLUSION AND FUTURE WORK

In this paper, we first propose a design and implementation method of streaming media live system based on multi-service node collaboration. The entropy weight method is used to dynamically determine the weight factors of each service node depends on the load information of service node. Then it points out the problem that the streaming media broadcast service based on rtmp protocol encounters in the mobile terminal. The solution proposed in this paper can solve this kind of problem. Finally, we write the system and test it via Microsoft Edge, thus we know the feasibility of the system is verified by the simulation and the system is implemented in the real network.

Our future work is to solve the large load of the coordinator. The coordinator needs to receive and return all index requests from user. If the coordinator does not respond within the specified time, the video broadcast is likely to be interrupted, resulting in a poor user experience. To this end, we will follow-up work on the coordinator is also based on multi-service node collaborative architecture to build, so that both reduce the pressure on the index server, without affecting the overall system's main logic.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (No. 61462014),the outstanding talents of science and technology education special fund project of Guizhou province governor under Grant (NO. 2011-34) (qianshengzhuanhezi), Natural Science Foundation of Guizhou Science & Technology Department under Grant (NO.2011-2201).

REFERENCES

[1] Jian Huang, DongMei Wu and XiaoPei Liu, "Implementation of the RTMP server based on embedded system," Computer Science and Information Processing (CSIP), 2012 International Conference on, Xi'an, Shaanxi, 2012, pp. 160-162.

[2] HU Guoqiang, ZHOU Zhaoyong, XIN Zhaoxia. Design and implementation of open source live system based on SRS. Modern Electronics Technique, 2016, pp. 36-39+43.

[3] P. Zhao, J. Li, J. Xi and X. Gou, "A Mobile Real-Time Video System Using RTMP," Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on, Mathura, 2012, pp. 61-64.

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

[5] H. T. Le, H. N. Nguyen, N. Pham Ngoc, A. T. Pham, H. Le Minh and T. C. Thang, "Quality-driven bitrate adaptation method for HTTP live-streaming," 2015 IEEE International Conference on Communication Workshop (ICCW), London, 2015, pp. 1771-1776.

[6] Pantos R,May E W.HTTP Live Streaming draft-pantos-httplive-streaming-20. https://tools.ietf.org/html/draft-pantos-http-live-streaming -16 . 2015

[7] T. C. Thang, H. T. Le, A. T. Pham and Y. M. Ro, "An Evaluation of Bitrate Adaptation Methods for HTTP Live Streaming," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 4, pp. 693-705, April 2014.

[8] WANG Rui,YANG Jie,TANG Ding. Research and implementation of video monitoring system based on Android and embedded Web server. International Electronic Elements, 2016, pp. 191-193.

[9] ZHANG Yongqiang, YU Boping, et al. Design and Implementation of a Campus Network TV System Supported Mobile Devices. Journal of Wuhan University (Natural Science Edition). 2012, pp. 365-370.

[10] A. Aloman, A. I. Ispas, P. Ciotirnae, R. Sanchez-Iborra and M. D. Cano, "Performance Evaluation of Video Streaming Using MPEG DASH, RTSP, and RTMP in Mobile Networks," 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC), Munich, 2015, pp. 144-151.

[11] P. Chakraborty, S. Dev and R. H. Naganur, "Dynamic HTTP Live Streaming Method for Live Feeds," 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, 2015, pp. 1394-1398.

[12] XU Xiangnan, FU Kun, LUO Shuzhen, et al.Redundancy Optimization in Live Streaming Transmission System. Video Engineering, 2014, pp. 61-64.

[13] LI Yunfei, XIE Weikai, et al. Optimization on behavior of HTTP cache server in live HTTP streaming system. Computer Engineering and Applications, 2012, 48(10), pp. 68-74.

[14] LUO Shuzhen, GENG Hengshan, et al. Research and Improvement of Streaming Media Live System Based on HLS Protocol. Journal of Zhengzhou University(Engineering Science), 2014, pp. 36-39.

[15] Yang Xue-Qiang, Li Wen-jun, Yue Yong.Comprehensive evaluation method to determine index weight[J].Journal of Academy of Armored Force Engineering, 2015:101-105. (in Chinese)

[16] N. Jing and J. Wang.Evaluate the International Talents in Professional Social Networks Using the Entropy Weight Method[J].Computational Science and Computational Intelligence(CSCI), 2015:220-225.

[17] N. Hongzhan and Y. Ting.Comprehensive evaluation for transmission network planning scheme based on entropy weight method and VIKOR method[J].Mechatronic Science,Electric Engineering and Computer (MEC), 2011:278-281.

[18] D. Ke, L. Pan and Z. Hui-ming.Comprehensive Fuzzy Evaluation for Power Transmission Network Planning Based on Entropy Weight Method[J].Intelligent Computation Technology and Automation, 2009,2:676-680.

[19] Qi Wang, Chong Wu, Yang Sun.Evaluating corporate social responsibility of airlines using entropy weight and grey relation analysis[J].Journal of Air Transport Management, 2015:55-62.

[20] Asadulla Khan Zaki and Indiramma M., "A novel redis security extension for NoSQL database using authentication and encryption," Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on, Coimbatore, 2015, pp. 1-6.