Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 12, No. 1, 2024

Research and Application of Mobile Video Content Placement Based on Edge Computing

Congling Yan

Hunan Institute of Humanities, Science and Technology, Loudi, China

Abstract: In the era of the rapid development of computer technology and Internet, the traditional cloud server architecture and solutions have been unable to meet the needs of the huge Internet users, with the rapid development of mobile video, traditional server resilience, server cost and user experience has been unable to meet the needs of users and enterprises. Under this requirement, the edge computing platform based on kubernetes is emerged to solve the above problems. This paper first analyzes the development process and characteristics of kubernetes, and further discusses the application of mobile video cache strategy on kubernetes.

Keywords: Edge Computing; Internet; Kubernetes; Co-caching; Cache Replacement Strategy.

1. Introduction

Mobile video, as the hottest topic at present, from the 2G and 3G era at the beginning of the 21st century to the 5G era, the Internet has also changed from traditional text pictures to the application of video and other technologies. With the continuous improvement of network quality, user experience has become a very important topic. In this context, how to efficiently store and manage mobile video and how to apply kubernetes and mobile video applications has become an indispensable technical problem. On the kubernetes-based edge computing platform, mobile video placement and user experience can be greatly improved. Based on this, this paper first expounds the introduction, characteristics and application of kubernetes, edge computing and cache strategy. With this approach, it analyzes the application of edge computing, kubernetes and cache strategy to mobile video, and puts forward reasonable assumptions and verification to promote the development and application of edge computing in mobile video.

Storage strategy for mobile video placement via edge computing, We were able to reduce the pressure on the server, Improve the resilience of cloud servers, Transfer of multiple network shocks to the edge servers, Can effectively reduce the server pressure, And is able to maximize the efficiency, Through the edge server, Everyone can access their own resources faster and with less delay, In the past, user access to the same cloud server, so, Cloud servers are under high pressure, With the Internet or other problems, Maintenance time and maintenance costs are very high, And different user experience in different regions, Now that edge computing can be well resolved, Users will first access the nearby resource, Can solve the problem of time delay very well, And users can not only access resources that match their own telecommunications resources, Such as mobile access to mobile, China Unicom Access to China Unicom, Telecom Access to Telecom, This can further improve the quality of the network.

2. Edge Calculation and Kubernetes

2.1. Concept and Features of Edge Calculation and Kubernetes

Edge computing is essentially a content distribution network, first dating back to 1998. Content distribution network is now called CDN, through this technology, deployed cache server, thus, user access resources without source station, but access to the nearest cache server, through the cache server scheduling system, network shunt, reduce the network congestion, and through the corresponding cache strategy, improve the user's cache, and the user experience, is the user access response speed.

Kubernetes Is a Google open-source container orchestration technology used to manage cloud platforms. It can conduct a distributed cluster deployment on the cloud platform, with high availability, high scalability, etc. kubernetes can make the deployment of container technology easier and more efficient, with less resource footprint, fast deployment and startup speed, low loss, and high portability. Kubernetes With the master node (Master) and the working (Node) node, the daemon process services running on the master node are kube-apiserver, kube-scheduler, kube-controller-manager, etcd, Pod. The working node is the site of the Node. Kubenetes Operating time of docker, rkt, etc. The kubenetes component executed on the Node is kubelet, kube-proxy.

kubernetes Is a container layout tool on the market, container is a package contains the environment and all dependent on, container can be deployed in any environment, kubernetes is an open source project, it is responsible for in the mass server environment management container group (pod) extension, replication, health, and solve the problem of pod starting, load balancing. As long as the application can be packed into the container, kubernetes Can be done on the physical server, virtual machine, Cloud environment and other security start, kubernetes You can also switch seamlessly switched in the cloud environment, kubernetes Not saturated at the discovery node, The program will assign the pod, Can save money, Efficient utilization of processors, memory, If one node is going down, kubernetes Will automatically in the other node ship running in this node pod,

kubernetes Also has the ability to contract contract, In, at the too high load, Pods are automatically created when they is insufficient, When the load decreases, kubernetes The deployed applications will always run safely, If a fault occurs, kubernetes Will automatically assign applications to healthy pods, kubernetes Make it easy and reliable for us to deploy, start, and migrate our applications. kubernetes Includes the Master node and the Node node. Master Nodes running Daemon services are kube-apiserver, kube-scheduler, kube-controller-manager, etcd, Pod network Node nodes is where Pod runs Kubenetes supports docker rkt and other container Runtime. The kubenetes components running on Node are kubelet, kube-xy.

Kube-apiserver gave HTTP / HTTPS restful api, or kubernetes API. The network server is the special tool (CLI and UI) for the front socket of the kubernetes cluster, which can manage the various network resources of the method Kubernetes cluster.

Kube-scheduler: Take the decision on which node to put the pod on. When scheduling, the scheduler will fully consider the current load of each node in the cluster topology and its requirements for high availability, performance and data affinity.

Controller Manager: Responsible for managing various resources of Cluster to ensure that the resources are in the expected state Controller Manager eplication controller endpoints controller, namespace controller servicaccounts controller and other components of namespace controller to manage Namespace resources.

Etcd: etcd stores the equipment information of kubernets clusters and the condition information of various natural resources. When the data information changes, etcd quickly notifies the kubernetes about the parts.

The Pod network: The Pod networks can communicate with each other. Kubernetes The cluster must deploy the Pod network. flanner, This is one of the options, and it is also the default schedule of the kubernetes official network.

The Node node service: Kubelet is a node agent. When the production scheduler process decision runs the Pod on a node, it pushes the special configuration information (image, volume, etc.) to the kubelet of the node. With this information, kubelet kublet reports the operation to the server. Kube-proxy logically indicates several pods. How to share the Pod according to the external requirements of the Pod service? That's what the kube-proxy does. Each node will run the Kube-porxy service and share the TCP / UDP data flow analysis of the browsing service to the following container. If there are several groups, the cube-proxy will complete the load balancing of the web services.

Master Is a management node, can also serve as a word node, sometimes scheduler may be deployed to the node node, this is for the sake of high availability, so that each node can be master or node node, so as to have more solutions when problems occur, to avoid the service outage.

2.2. The Concept of a Caching Strategy

Classification of caching policies:

Time based on access: the algorithm decides the change target according to the access time mechanism cache sequence of each cache item. for instance LRU.

Based on the access frequency: The algorithm caches the mechanism based on the access frequency of the cache item. For as LFU, LRU 2,2Q, LIRS.

Visit time and frequency: consider the time and frequency of the visit. Therefore, when the data information mode changes, the cache strategy still has good characteristics. Such as FBR, LRUF, and ALRFU. Most of these algorithms have an adjustable or responsive main parameter, and the adjustment of this main parameter prompts the cache strategy to obtain an equilibrium between the access time and the access frequency.

Based on access mode: there are some uses of clear data information access characteristics, and then form the corresponding cache strategy. For example, the A & L caching strategy developed by the dedicated VoD system adapts to the SARS strategy with both random and sequential access modes.

There are five typical algorithms for the CDN cache algorithm:

1), the "Recent least use" cache algorithm

2). The "minimum frequency use" cache algorithm

3), the "score-based factor" caching algorithm

4). The "Block level" cache algorithm

Based on the above algorithm, we can improve a good cache hit rate, but also save storage costs.

3. Acquisition and Analysis of Mobile Video Data

3.1. Mobile Video Data Acquisition

3.1.1. Data Acquisition Process

This section takes the mobile video platform TikTok as an example to collect the data. Acquisition platform: douying.com, acquisition system: windows10. This section collects data for mobile video by simulating the user browsing the video.

Figure 1. Collection homepage of the video platform

Through reverse analysis of web page request, get the title of the video, video likes, video views, video comments, video forwarding, video upload address, real video stream, on this basis, I wrote a js script, by simulating the user click, video switch, through real video download address, download the video switch, and save the file name for this information as shown in figure:

Other information about the video is kept in the csv file for subsequent analysis.

Due to the TikTok video recommendation, different video push will be carried out along with the viewing habits of users 'accounts, users' attention, comments, and users' registration places. In this way, we registered some new accounts through Tencent Cloud or other servers in different regions, by installing windows systems, installing browsers, and using scripts. Through these new blank accounts, avoid using these accounts to interact with other users and videos, so as to ensure that the account data is clean during the measurement process. By constantly watching the video, to get the data we

need.

240元12个菜2个汤，山东大席小王子，人帅菜硬气..@28.6w@2.1w@小钉探美食.mp4
海南万宁结婚流水席，1桌13个菜摆50桌，厨师称烧柴火是地方@1.0w@625@麦总去哪吃.mp4
#薛之谦苏州演唱会大串烧！！如果你也喜欢 如果有幸刷到 如果@3.5w@3297@老徐徐徐.mp4
#薛之谦广州演唱会 全场最高能的15分钟全场大合唱了 #万人@1.4w@1035@MYK46ISYOU.mp4
#万能实习生 探索五百种赚钱方法，今天是接受全网挑战卖万物的@4.9w@2537@万能实习生.mp4
30位逆袭翻身的弟弟：打弟弟要趁早，不然长大后你只有挨打的份@1203@23@小七仔爱混剪.mp4
方便面滑板2.0 之前发的因音乐版权被下了，只能重新发一次，@187.5w@6.8w@Journey滑板店.mp4
ofo大败局【20分钟完整版】 #ofo #戴威@2.3w@63@财可夫斯基.mp4
#一定要看到最后@4.3w@614@芷萱mm.mp4
#我的乡村生活 #记录真实生活 #大山歌唱@257.7w@14.7w@张同学..mp4
7倍速视频告诉你西大有多大#西南大学 #校园@4.4w@1.4w@Renly.mp4
#我的乡村生活 #记录真实生活 #新农人计划2021@177.7w@10.8w@张同学..mp4
10年黄河老鲤鱼一米多长，胡须长的就差跳龙门了，摄影师激动的@1.2w@1013@农人刚子.mp4
我是马思唯. #马思唯 #livehouse #蔚然花海音乐@19.2w@2.4w@水星.mp4
盘点90后父母带娃搞笑名场面，还想用我小时候的招数，来对付我@19.1w@2.7w@迷惑侠.mp4
#新农人计划2021 #我的乡村生活 #记录真实生活@198.4w@11.5w@张同学..mp4
100个无限子弹的特种兵对阵60000个古代战士，谁能胜？@2.3w@9003@五一电影院.mp4
全网生活中百年难遇的爆笑瞬间：女孩去纹身老板却要给她纹前面？@4281@163@小黑爆笑社.mp4
盘点网购翻车名场面,假一赔三直接发四双鞋,商家你的良心不会痛@10.1w@5488@驴儿搞笑视频.mp4
闽江学院版"YesOK" 2.0#校园生活 #运动会@174.8w@10.0w@闽江学院.mp4
北京的城市逻辑，哪些央企会走？ #北京 #央企 #大湾区 #@15.6w@83@小民哥大冒险.mp4
全网那些"杀人诛心"社死现场！果真是伤害性不大，侮辱性极强@13.9w@4105@橘子撩娱.mp4
03年宋美龄在睡梦中去世，遗体被毛毯裹着抬出，只留一座无字墓@1321@0@啊哈哈，嗬！.mp4

Figure 2. collects the video data

3.1.2. Data Analysis

According to the video data we obtained, we obtained a total of 1341 videos and 1256 users, among which 279 videos were uploaded and targeted. The number of video likes varies from 100 to 200w, and the number of comments varies from 0 to 10w.

3.2. Analysis of the Number of Likes, Comments and Page Views of the Videos

Through the video release time distance now thumb up and views analysis, can know that video in just released thumb up and views upload slower, in 1-3 days, video likes and views will enter a rapid growth moment, after the video over time, thumb up and playback will gradually decline, after the 10th day, arrived at the bottom of the slow growth.

At the same time, it is found that through localized videos, users are interested in videos in the same region, and videos in the same city are generally only spread within the same city, rarely to other regions.

By analyzing the content of the video comments, the interaction between the users and the video creators is different. By crawling the video content, we found that in the case of the video creator often replying to users, the average number of video comments of the video creator is significantly more than that in the case of the same fan, and the number of videos that is not often in batches.

At the same time, through the TikTok search hot list, it is found that if the videos uploaded by users match the corresponding hot spots on the day, the number of videos and comments uploaded by users will be significantly improved, which is not the hot spots.

As a result, we can find that TikTok short video has many different characteristics compared with traditional video, users prefer to watch hot video, with the trend of The Times and follow up, users prefer to watch released more new video, users prefer to watch the video in the same region, users prefer to watch video bloggers more active video.

4. Architecture Design of the Mobile Video Cache System

A reasonable system needs a reasonable planning for the system. In this paper makes a reasonable configuration from the selection of the server system, the configuration of the server cluster, and expounds the whole deployment of the system and the construction process of the system. This system adopts the k8s system for the clustering of all servers.

4.1. Introduction of the System Architecture

The system consists of 9 servers cluster, and other servers build source station services for CDN cache effect test.

First of all, in terms of server configuration, we adopt 2H4G Master node uses 2H4G, other Node nodes use 1H2G, 2 H 4 G, 4H4G configuration, only 4H4G uses non-fixed public network ip and ip restricting some ports.as illustrated in following figure.

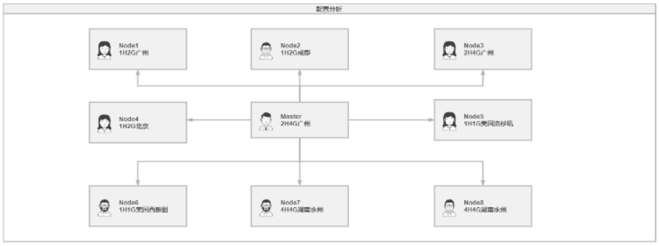

Figure 3. System Architecture diagram

Master The node runs the management side of the Goedge. While the remaining Node node, runs the Goedge through the Docker container.

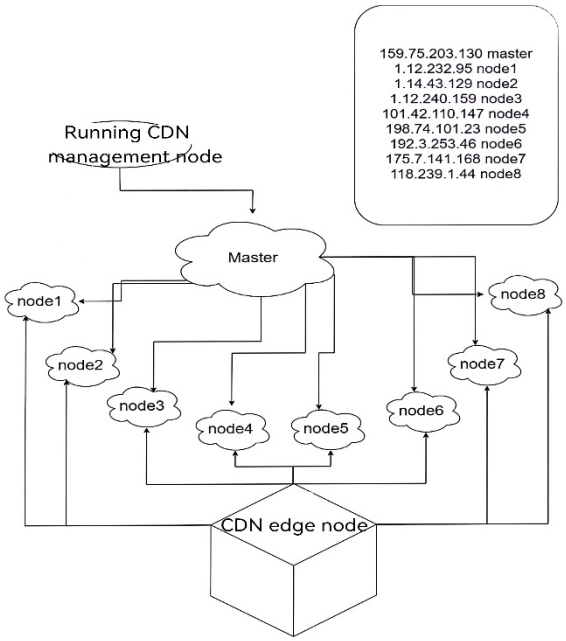

Figure 4. Function diagram of the system architecture

4.2. Production of Caching Platform Docker Mirror

This section writes reasonable Dockerfile files by analyzing the functions of the Goedge system and uploads them to the Dockerhub warehouse. This chapter is divided into three documents, Dockerfile is responsible for the production of mirrors, install. The sh is responsible for the mirror packaging process of the goedge system installation, run. And sh is responsible for the goedge system run.

Put these three files in the same directory of the linux system, through the docker built-t mirror name: version number. Can complete the production of the docker mirror. Through docker images, you can view the docker image just created. After creating the dockerhub warehouse account, log in through docker login- -username= user name. After logging

in, pass docker tag edge lcy0828 / goedgel: latest

Change the tag and upload it to the warehouse via docker push lcy0828 / goedgel: latest. Now you can directly use the docker pull lcy0828 / goedgel: latest to pull off the mirror for direct use.

4.3. Production of Kubernetes Pod Components

The Kubernetes pod component needs to be run based on k8s and docker images. The K8s pod component needs to rely on the yaml file to run through the deployment component.

This document is adapted using the yaml file of nginx, and adds different options according to the container to meet our requirements. Under the master node, passing through the kubectl apply-f goedge. The yaml run the pod

[root@master ~]# kubectl apply -f goedge.yaml

deployment.apps/edgecdn unchanged

Of course, this is the result of me and the run, and you can view the results through kubectl get-o wide,

You can see eight pods running.

5. Designs a Dynamic Cache Replacement Strategy

5.1. Dynamic Price Comparison Cache Replacement Strategy based on Video Popularity

For our caching system for mobile video, we first need to build a CDN agent system by using the public network server. Through this system, we can conduct the agent access to resources and the cache of resources. In this process, we need to use enough servers to complete the building of our cluster system and the need to use enough storage space for the servers, to deploy our resources, and resource services. Based on such a factor, in the process of continuous proxy and resource caching, the server configuration needs to be constantly upgraded, and there is no upper limit. Without the cdn proxy system, the bandwidth of our server will not be able to meet the demand of the peak network. Therefore, we can actively cache the required content to the edge server in advance, or migrate small computing tasks to the edge server, so as to reduce the network return traffic through the core network, and improve the response rate and service efficiency. In the above system, the basic process of the service user is that when the user initiates a request to the local edge server, access the other edge server in the collaboration area, access the remote data center through the core network, the data center transfers the content to the local edge server, and the edge server delivers the content to the relevant user and updates the local cache.

Through the analysis of mobile videos, this section analyzes that users are interested in the videos with high number of thumb up and hot spots corresponding to the current ones. Therefore, we design a dynamic cache replacement strategy algorithm.

Because we through the reverse engineering collected the mobile short video platform TikTok data, we need to through the analysis of the data design an algorithm based on the popularity, through 14 sky video thumb up, comments, and the length of video for a price calculation, design an algorithm, according to the video upload time to design a price, from 7 to 14 days to gradually reduce the price, the video since upload quantity price design, according to the real-time hot spots to related video price, etc., on the basis of calculate the price of all video and sort, take the first part of the cache.

The final price algorithm is that the final price is K, the uploaded video is T days ago, the video size is M, the video like quantity is J, and the video comment volume is L.

$$K=((T-14)*(A)+(M-3)*(B)+(J/1000)*C+(L/1000)*D)/(A+B+C+D)$$

By setting the proportion of A, B, C and D, it eventually becomes a reasonable cache replacement strategy that can be based on the current video content and the content and popularity of the video.

5.2. Design the Edge Collaborative Caching Strategy based on Time and Space

Due to the increasing number of videos, cache rate will be according to the current region will be different, at the same time there will be some waste of resources, for the user, access to the current region and the adjacent area of the actual experience is not big, and to access the edge nodes and access the source station experience than before will have a big difference. Therefore, we need to have limited access to the adjacent edge nodes when the cache misses. If the adjacent edge nodes have the video content we want, we will take it from the adjacent edge nodes. For example, Xinjiang has a vast territory, so it needs to cost too much to cover Xinjiang well, while the population in Xinjiang is very rare. We can build nodes in the surrounding areas of Xinjiang, provide a small coverage in Xinjiang, and the limited requests from Xinjiang can be allocated to adjacent areas.

For this purpose. We designed an edge collaborative caching strategy designed according to the upload area and the user's location.

6. Kubernetes Test and the CDN Functional Test

This section tests the stability of the system by testing the clustering of Kubernetes and the CDN function

6.1. Kubernetes Test

6.1.1. Kubernetes Cluster Test

This test takes sampling test. Since there are 8 working nodes in this cluster, this test takes one node from three different operators for the switch test, testing whether the cluster can be restored after disconnect.

6.1.2. Kubernetes High Usability Test

This section mainly tests whether the service runs on another device when a device is down.

6.1.3. Kubernetes Functional Test

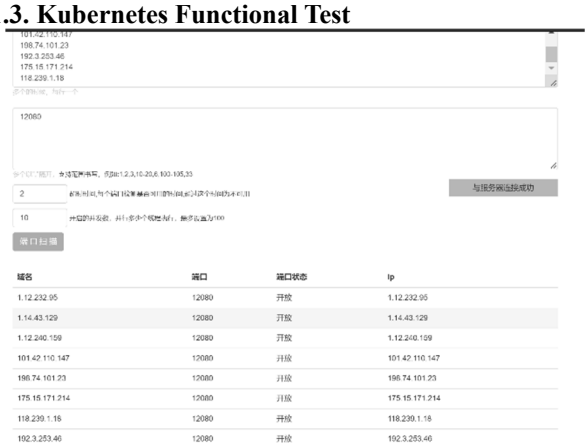

Figure 5. Port test

This subsection is tested by examining the connectivity of all the pod components.

Use the tool PostJson V2.0 to test the port, and the test results are universal connection.

6.2. CDN Agent Test

This section tests the success between the CDN system agent before and after the agent.

6.2.1. Proxy Function Test

Generate a random and unique share address in the source site http: / / video.lcy.pub/s/5Ycy

Figure 6. Service test of the source station

The agent is followed by accessing the http: / / cdnvideo.lcy. Test for pub / s / 5 Yc

Figure 7. The proxy service test

Successful test, with successful access

6.2.2. Stability Test

This is the continuous ping test of the source station

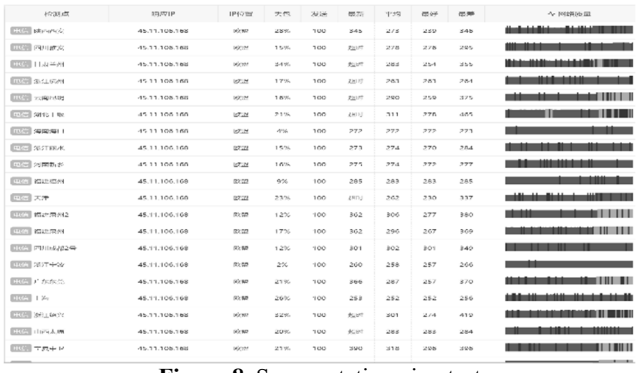

Figure 8. Source station ping test

This is a continuous ping test after using the cdn proxy for the same time

Through comparison, it can be seen to see that the original source server (source station) is located in Amsterdam, the capital of the Netherlands, and China will have a large area of packet loss problem when accessing the server, and after using the agent, the problem can be obviously solved in static resources.

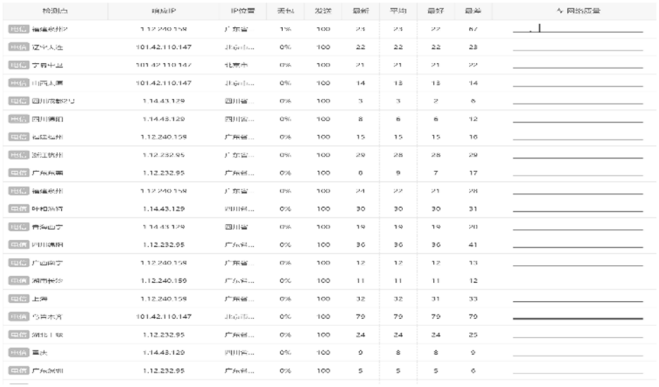

Figure 9. Service ping test

6.2.3. Response Speed Test

This subsection uses the itdog website as our test tool, covering the vast majority of the country as well as different lines.

The following is true for direct access to the source site:

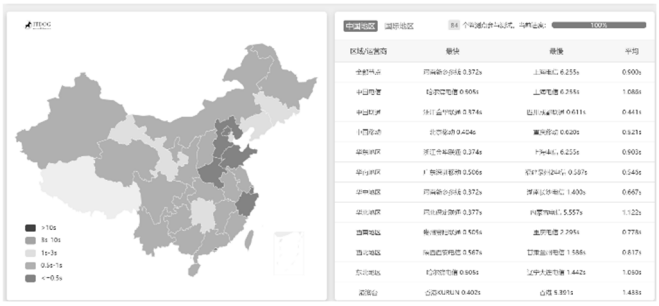

Figure 10. Test diagram of the source station

The following is the CDN proxy:

Figure 11. The cdn proxy test diagram

It is obvious that both the overall access speed and average access speed in all regions of the country, the cdn is faster than not. Of course, one of the main functions of cdn is to reduce the pressure on the source server.

6.3. Simulation test of cache replacement strategy

This subsection tests the replacement strategy algorithm based on video popularity. The simulation test program running in my personal laptop, by writing the GO language program, set reasonable value to test, the test process, set video number of 1000, set the cache space size of 100, through to 100000000 access, by comparing with random cache and FIFO algorithm, by measuring the data for many times, and to the average. The following statistical figures are obtained:

As can be seen from the picture, the cache hit rate is better than the FIFO algorithm and the random algorithm strategy with the increasing number of visits. After increasing the cache of the server, we can see that the proposed algorithm is

better than the FIFO algorithm and the stochastic algorithm strategy.

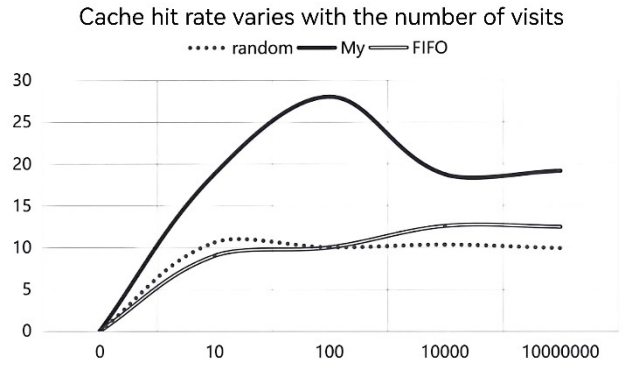

Figure 12. Details of different algorithm in cache hit rate

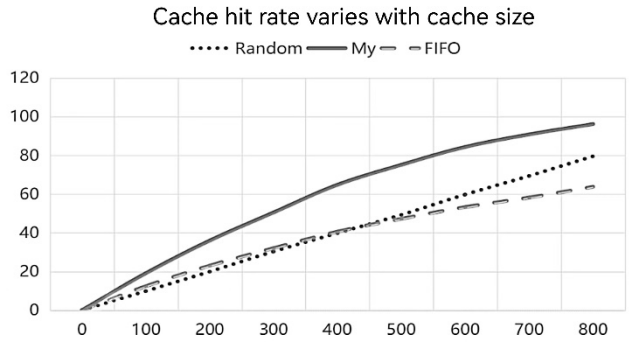

Figure 13. Change of cache hit rate with cache size

And collaborative cache strategy based on time and space, can be tested through direct access, by monitoring the user to the edge of the node and the user to collaborative cache node or the user to the edge and the delay between cache, a test statistics, and compared with the user directly access the source, test site are Hunan Yongzhou, test data are the average of multiple data, can draw the following chart.

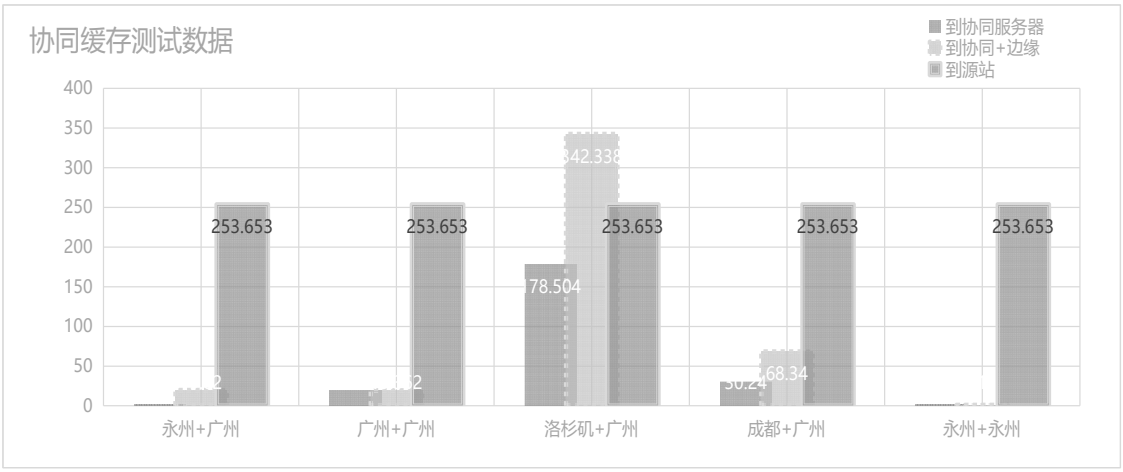

Figure 14. Collaborative cache test

It can be seen from the chart that the reasonable use of collaborative cache can greatly improve the speed of user experience, but the direct collocation of unreasonable edge nodes and collaborative cache nodes will have a negative effect. Therefore, our collaborative cache should be close to the geographical location of the edge server or have a lower delay.

7. Summary

Traditional cloud server to the user's model, has many disadvantages, users to access their resources, usually by accessing a cloud server, and at the same time a large number of users access, can lead to a significant decline in speed, at the same time also lead to a waste of resources, this mode, the cloud server needs great cpu computing power, as well as the large capacity of a hard disk space, to store all the user's resources, and relatively insecure, cloud server exposed in everyone's field of vision.

Edge computing has many advantages, the most important is that it can be any side equipment, whether router, base station, personal computer or mobile phone, it can exist as an edge server, previous access to resources through many paths to obtain resources, and through the edge calculation, you can access the base station resources, can greatly reduce unnecessary waste of resources, and can reduce network congestion, now even you want to get a resource, is through the edge of the same village server access.

Through the edge of the kubernetes computing and cache strategy, can reduce the cloud server pressure, improve the user experience, reduce the user brush video delay, and avoid network problems caused by service failure, improve the hit rate of cache and reduce the user to the edge of the server delay, and reduce the cost of cache the same content.

This paper begins with a detailed discussion of the kubernetes and caching strategies, Then make the Docker image by writing Dockerfile and shell scripts, Made to the pod by writing the yaml file, In a detailed description of the entire building process of kubernetes and goedge related services, Finally, by studying the cache replacement strategy, We have devised a rational algorithm, And, through our systematic and reasonable testing, Verify that our algorithm for improving the cache hit rate, And the advantage of improving the service response speed. Through reasonable testing to my views and the stability and system of the feasibility of a verification.

References

[1] Hu Yuhan. Mobile short video: measurement, analysis, and edge cache [D]. University of Science and Technology of China, 2021. DOI:10.27517/d.cnki.gzkju.2021.001680.

[2] Zhang Kaiyuan, GUI Xiaolin, Ren Dewang, Li Jing, Wu Jie, Ren Dongsheng. Review of computational migration and content caching in mobile edge networks [J]. Journal of Software, 2019,30(08):2491-2516.DOI: 10.13328/j. cnki.jos. 005861.

[3] Wang Haixia. Research on content Cache Optimization technology in mobile edge computing system [D]. Zhejiang University, 2019.

[4] Ge Zhicheng, Xu Ke, Chen Liang, Li Tong, Yao Long, Shen Meng. A hierarchical collaborative caching mechanism for mobile content distribution networks [J]. Journal of Computer Science, 2018,41 (12): 2769-2786.

[5] Wang Ruichao. Research on edge-end collaborative caching algorithm for mobile video service [D]. Beijing University of Posts and Telecommunications, 2021.DOI:10. 26969/d. cnki. gbydu. 2021.002177.

[6] Li Ruixiang, Mao Yingchi, Hao Shuai. Cache management method based on approximate matching [J]. Computer Science, 2021,48 (01): 96-102.

[7] B.Jedari, G. Premsankar, G. Illahi, M. Di Francesco, A.Mehrabi and A. Ylä-Jääski, "Erratum Erratum to “Video Caching, Analytics, and Delivery at the Wireless Edge: A Survey and Future Directions”," in IEEE Communications Surveys & Tutorials, vol.23, no.2, pp.1421-1424, Secondquarter 2021, doi: 10.1109/COMST.2021.3063864.

[8] Li Chunlin, Liu Jun, Zhang Qingchuan, Luo Youlong. Efficient cooperative cache management for latency-aware data intelligent processing in edge environment[J].Future Generation Computer Systems,2021(prepublish).

[9] Li Chunlin, Zhang Yong,Song Mingyang,Yan Xin,Luo Youlong. An optimized content caching strategy for video stream in edge-cloud environment[J].Journal of Network and Computer Applications,2021(prepublish).

[10] L.Gu, D.Zeng, J.Hu, B.Li and H.Jin, "Layer Aware Microservice Placement and Request Scheduling at the Edge," IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp.1-9, doi: 10.1109/ INFOCOM42981. 2021.9488779.

[11] S.Pasteris, S.Wang, M.Herbster and T.He, "Service Placement with Provable Guarantees in Heterogeneous Edge Computing Systems," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp.514-522, doi: 10.1109/ INFOCOM. 2019.8737449.

[12] V.Farhadi et al., "Service Placement and Request Scheduling for Data-Intensive Applications in Edge Clouds," in IEEE/ACM Transactions on Networking, vol.29, no.2, pp.779-792, April 2021, doi: 10.1109/TNET.2020.3048613.

[13] Y. Liang et al., "Interaction-Oriented Service Entity Placement in Edge Computing," in IEEE Transactions on Mobile Computing, vol.20, no.3, pp.1064-1075, 1 March 2021, doi: 10.1109/TMC.2019.2952097.

[14] A hierarchical collaborative caching mechanism for mobile content distribution networks [J]. Ge Zhicheng, Xu Ke, Chen Liang, Li Tong, Yao Long, Shen Meng. Journal of Computer Science. 2018(12).

[15] Gold wei. Design and implementation of Docker distributed container automatic operation and maintenance system based on K8S [D]. South-Center University for Nationalities.2018.

[16] Gao Jixu. Research on the unloading strategy of collaborative computing tasks based on edge computing [D]. Nanjing University of Posts and Telecommunications.2021.