Research and Implementation of Edge Computing in Web AR

Research and Implementation of Edge Computing in Web AR

Haoran Yan[1], Xiuquan Qiao[1,*]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding author e-mail: qiaoxq@bupt.edu.cn

Abstract. Cloud computing, which includes a wide variety of cloud services, has experienced explosive growth in recent years and cloud servers are replacing traditional physical servers on a global scale. However, some emerging businesses such as AR (Augmented Reality) have made the traditional centralized cloud computing expose some shortcomings. For example, the actual physical or network distance between clients and clouds is too large and the growth rate of bandwidth resources in the cloud is far behind the growth rate of data, which makes cloud computing unadaptable for the requirements of AR computing that has high bandwidth occupancy as well as low latency. To handle these problems, this paper introduces and implements a novel idea of edge computing in the field of Web AR. Meanwhile, experiments demonstrate that edge servers have the capability of real-time complex computing with high efficiency and stability.

1. Introduction

Augmented Reality, a new technology of inheriting real world information and virtual world information [1], has a strong sensory experience and broad application prospects in fields of education, medical, advertising, entertainment, etc.

In defiance of this enormous market volume most of the AR prototypes were not able to evolve into merchantable products [2]. This is mainly because there are various drawbacks in the current technological forms for providing AR services:

- Dedicated AR equipment: A heavyweight solution. Bimber and Raskar [3] divided it into head-mounted, hand-held and space projections depending on the application scenario. The optimization of dedicated equipment makes the overall AR experience excellent, but there are two obstacles. The first one is high manufacturing cost, for example, Microsoft's HoloLens sells almost $3000. The second one is poor portability. Although google has tried to make some improvements, Google Glasses was failed to run the demanding AR algorithm due to the small size of the device.
- Relying on smartphone: Due to the development of the chip and storage industry, the computing and storage capabilities of the smartphone terminal have also improved dramatically. There are two methods to implement AR services on cellphone: native app and Web app. Since most of the AR functions can be run through the smartphone itself, the well-configured flagship model guarantees a certain real-time performance and common user experience by native app. However, it is difficult to mass-spread since you need to install corresponding app. The other is Web app, which is a pure front-end Web AR, used WebRTC to obtain camera images, and used AR.js, JSFeat and other JavaScript libraries to provide

limited AR functions. Web APP greatly reduces user cost, but the computing power of the browser itself makes the AR service capability greatly degraded.

- Relying on cloud computing: This is also the solution adopted by most of the large domestic manufacturers, such as Alipay's Sweeping Fu and QQ's AR Olympic Torch. Although cloud provides an efficient computing platform, the current growth rate of network bandwidth is far behind the growth rate of data and the decline of network bandwidth cost is much slower than the cost of hardware resources such as CPU and memory [4]. Due to the geographical dispersion of users and the complex network environment, even if a large amount of money is invested in bandwidth, it is difficult to ensure the lower transmission delay of the network. Therefore, the traditional cloud computing model needs to solve the two bottlenecks of bandwidth and delay [5].

In order to solve the above two problems, this paper introduces edge computing, a computing mode that is generally understood to mean that both computing and storage are close to the edge of the network, therefore ending users use it at the edge of the network, into traditional cloud computing that form a new architecture with powerful and scalable computing capability. Therefore, it completes a Web AR framework with cloud, edge and end's AR computing tasks in a distributed collaborative environment.

Compared with former cloud computing, edge computing can better support AR computing scenarios: (1) Cisco pointed out in the Global Cloud Index that global devices will generate 600ZB data in 2020 [6], and 90% of which are temporary data just like AR scenarios. A large amount of temporary data can be stored at the edge nodes to alleviate the pressure on the cloud bandwidth. (2) High delay, strong jitter and low data transmission rate caused by the unstable links and routes in the complex network environment, affect the responsiveness of cloud services [7]. The edge-side is closer to the user-side in both geographical distance and network distance, which ensures lower latency and reduces network jitter, making edge calculation more useful and responsiveness stronger [8]. (3) The images involved in the AR computing, such as face data, which belongs to the user's private data. This part of the data is stored at the edge, which reduces the possibility of privacy leakage.

This paper relies on our lab's Web AR service platform to complete image acquisition, recognition, 3D model loading and rendering as a test scenario after using edge computing nodes. Detailed test and summary of relevant performance indicators under 2.4 GHz shows great improvements.

2. Related Work

Edge computing currently doesn't have unified and strict definition. Mahadev Satyanarayanan [9] defined it as: "Edge computing is a new computing model that deploys computing and storage resources (e.g. Cloudlets, micro data centers, or fog nodes) to networks closer to mobile devices or sensors."

Currently, the field of edge computing research involves fog computing, Cloudlet, and mobile edge computing (MEC) in the upcoming 5G era.

2.1. Fog Computing

Cisco proposed fog computing in 2012 [10], which is actually an extension of the cloud network structure from the perspective of cloud computing. Although it is very similar to edge computing, it is more concerned with the infrastructure level rather than the application-level, while AR scenarios have higher requirements for response time and perceived experience. The literature [11] introduces the concepts and application fields of fog calculation in detail.

2.2. Cloudlet

Cloudlet was proposed by Carnegie Mellon University in 2009 [12], deployed between the cloud and mobile terminals. As a middle layer, Cloudlet virtualizes edge resources to manage them through the OpenStack API. Meanwhile, Cloudlet provides technologies of service discovery and switch focused on the versatility of infrastructure layer, but not on optimization for AR scenarios

.

2.3. Mobile Edge Computing

With the advent of 5G, mobile edge computing [13] is a popular topic of research community nowadays, as well as a key technology for the development of 5G and a new programming paradigm for large-scale Internet of Things applications [14]. In addition to satisfying the basic access of the user, the 5G base station can also handle part of user's computing tasks. The introduction of the MEC server, together with the advantages of 5G in high bandwidth, low latency and scalability, can greatly reduce the cloud network congestion.

Because this paper focuses on the optimization of the edge computing tasks in the AR field, and does not conflict with 5G itself, after the large-scale commercial use of 5G in the future, it can greatly enhance the service experience of AR applications.

3. Architecture

The computing resources of the generalized network edge can include a series of devices such as mobile phones, PCs, base stations, WiFi APs, cameras, small computing centers, etc. This paper thinks that the mobile phone is a user-side equipment with certain computing power. So, we limit the edge computing resources to the edge computing nodes (edge servers) in edge network.

Considering the lightweight and universal requirements of AR applications, this paper discusses the most common application scenarios in Web AR, which can be abstracted for image acquisition, recognition matching, loading 3D models, etc.

Figure 1 shows the Web AR model in a traditional centralized cloud computing framework. The user's smartphone captures the image to be matched in the video stream through the camera, which can be certain Marker or Markerless (natural picture). The smartphone sends the picture to the cloud with request. After then the cloud uses large-scale computing resources to process the recognition task, and returns the corresponding 3D model stored in the cloud. Smartphone uploads and renders 3D model via browser's rendering technology. At this point, the current round of AR calculations is completed.

As can be seen from the above, with the spread of users, when the number of users increases in a large amount, there may appear the shortage of network bandwidth, the increase of delay, or the risk of user privacy leakage in the rectangle in Figure 1, make the traditional cloud computing model incompetent.

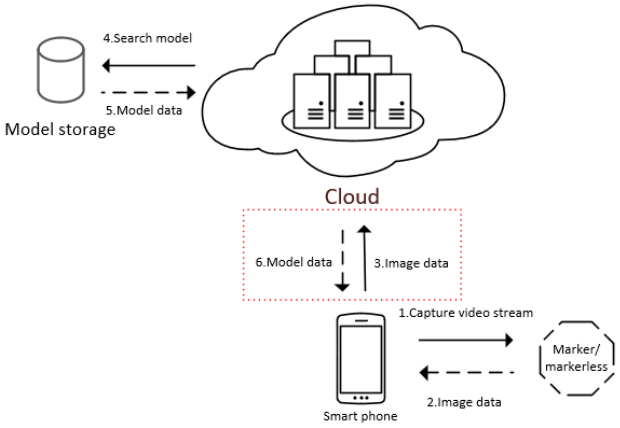

Figure 1. Centralized cloud computing model.

In response to the above problems, we introduced the edge computing layer on the client side and the cloud, which consists of a large number of small and low-cost servers. The specific network model is shown in Figure 2. Based on the geographical location of users, we divide the user service area into multiple edge areas. User's smartphones access the edge network through APs in a broad sense including cellular networks, WiFi, etc. The edge servers undertake AR computing tasks and 3D model's storage, so that the computing and storage are closer to the user side in physical distance and network distance. For each edge domain, it can be considered to be dependent, and unified by the cloud center. The specific process of a round of AR tasks is shown in Figure 3.

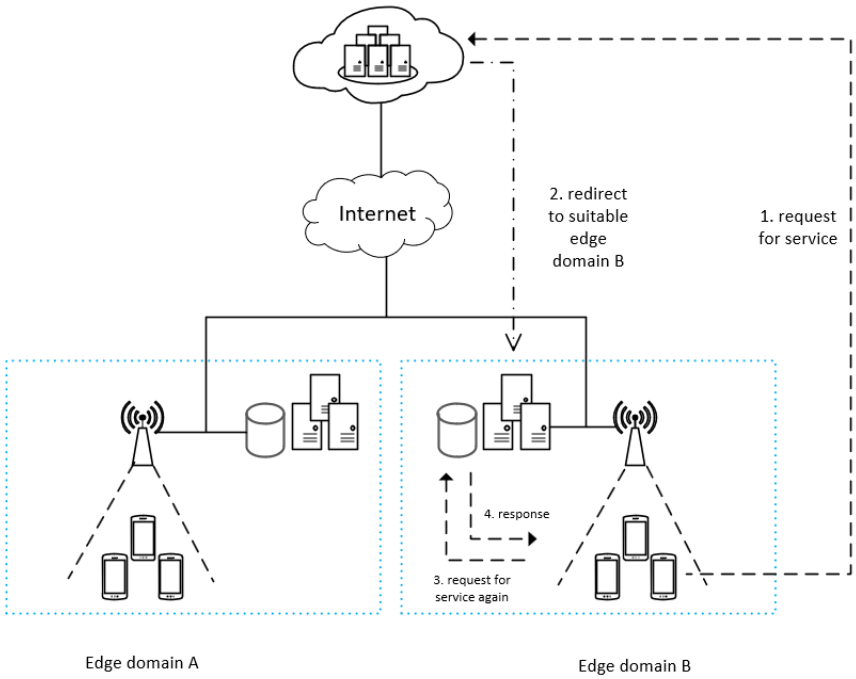

Figure 2. Edge computing model.

For a user, a complete AR service access is actually initiated from the cloud node first, for users don't know which edge domain they are in. The cloud node assumes the role of global load balancing. After receiving the user's request, it needs to select the most suitable edge computing node and redirect the request according to the outgoing IP address of the user. Subsequent computing tasks mainly occur in this relatively independent edge domain. According to the splitting of computing tasks, the independent AR engine and 3D model database are used to identify the target image. The next steps are similar to the above.

Each edge domain is equivalent to sharing the AR computing tasks that need to be transferred to the cloud end before. Usually, the edge domain where the user is located can be considered as a local area network, which can be composed of a large number of small servers and databases. According to our observation of the AR business, usually an AR service is also closely related to the corresponding location. Therefore, this computing model saves costs while improving capability of concurrency.

4. Implementation

Since AR computing tasks have high requirements in terms of latency and bandwidth, some specific optimizations need to be done in computing and storage. We choose more flexible and lightweight Docker container technology instead of virtual machine-based Cloudlet technology. Designed to deliver applications quickly, Docker is a cross-platform, portable and easy-to-use container solution. At the same time, in order to maximize the utilization of edge computing resources, we verified on the representative open source container cloud platform Kubernetes.

The basic operating unit of Kubernetes is Pod level-higher than native Docker, which avoids links between complex containers. Meanwhile, the Service defines a set of external access interfaces for Pods that provide the same service and guarantees the load balancing of the internal Pod of the service through the virtual IP.

According to the characteristics of the AR scenario, we divide the AR computing into front-end service, user DB service, AR core service, model storage service and other user-level services as Figure 3 shows.

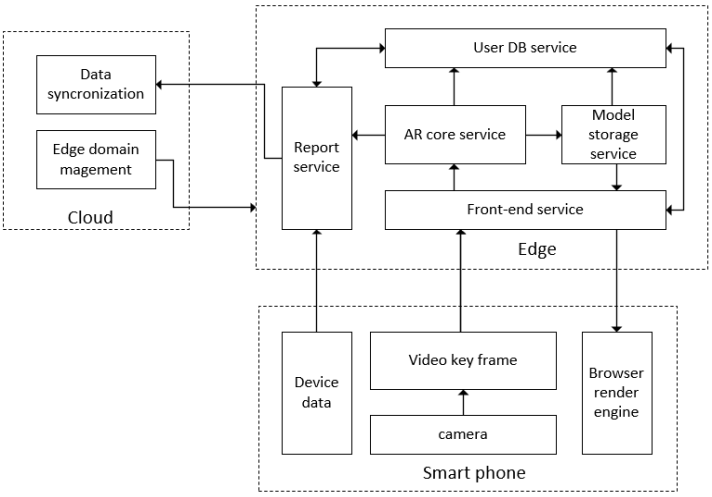

Figure 3. Association between edge services.

Front-end service: An external service, directly facing the user side and including AR activity UI and pages. The camera of user's device is picked up by WebRTC protocol. The key frame of the image is captured on the user side and transferred to front-end service.

AR core service: As an AR engine, it runs some kinds of algorithm that meets the needs of the business, which can be image recognition or tracking. In this test case, the surf algorithm based on feature point recognition is mainly used. As the core service of the entire edge computing, the number of Pods under Service can be adjusted according to the amount of traffic and the degree of computing load.

Model storage service: The fineness of the 3D model directly affects the sensory experience of the AR service, but the large models also cause high latency. The file formats of the 3d model data we use are fbx and obj. In order to minimize the transmission delay, we have already compressed at the time of storage, and maintain a k-v database to index the model file for reducing the search time.

User DB service: It is mainly used to store temporary business data generated by users at the edge and a traditional relational database can be chosen.

Report service: Due to the natural dispersion of the edge sides, it is not conducive to the control of the edge services. Therefore, the service is mainly used to periodically push the basic information of the entire edge network to the cloud, which includes the non-sensitive data generated by services and the monitoring data of k8s.

5. Experiment and Evaluation

To evaluate the performance of our edge computing system, we selected a total of 8 pictures for AR computing tasks and each picture was tested 10 times for average under the same condition. We used Alibaba Cloud as AR cloud computing service, deployed the edge service on the campus network together with the smart phone.

Table 1. AR computing latency comparing (time unit ms).

Average latency	1	2	3	4	5	6	7	8
Cloud	1311	1423	3142	3194	3679	4943	5455	5578
Edge	359	366	713	611	569	857	1083	947

As table 1 shows, after we sorted the experimental data in order by the cloud's latency, edge computing has excellent performance in time delay. We can split the overall delay into 3 parts as (1).

$$L = Size_p / BW_u + T_c + Size_m / BW_d \quad (1)$$

The difference of the time for AR computing T_c is small, even cloud is better. However, the time used to transfer data of picture and 3D model, controller by the upload and download bandwidth

respectively, is the key. The edge network's bandwidth can be 150Mbps or even more, while the cloud's is only 10Mbps for expensive price. Therefore, deploying the computing and storage to the edge node helps a lot for the experience of Web AR service.

6. Conclusion

In this paper, we proposed an edge computing architecture for Web AR scenario. Experimental data shows that with the architecture mentioned, it is possible to overcome the existing difficulties with the poor portability of dedicated AR equipment, the bad performance of pure front-end and the high cost of cloud computing. In addition, it brings advantages such as simplified deployment and secure storage of user's privacy data.

To achieve better results, the combination of 5G communication technology and edge computing is necessary. Higher bandwidth and lower latency can greatly enhance the service experience of not only AR applications but also IoT, car networking and other emerging applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants No. 61671081 and No. 61720106007, the Beijing Natural Science Foundation under Grant No. 4172042, and the 111 project (B18008), the Fundamental Research Funds for the Central Universities under Grant No. 2018XKJC01.

References

[1] KARHU A, HEIKKINEN A, KOSKELA T, Towards Augmented Reality Applications in A Mobile Web Context[C]// International Conference on Next Generation Mobile Apps, Services and Technologies, USA: IEEE, 2014:1-6. DOI: 10.1109/NGMAST.2014.36
[2] Michael Schneider, Jason Rambach, Didier Stricker. "Augmented reality based on edge computing using the example of remote live support", 2017 IEEE International Conference on Industrial Technology (ICIT), 2017
[3] Bimber O, Raskar R, Modern approaches to augmented reality. In: Proceedings of ACM SIGGRAPH 2006 Courses, Boston: ACM, 2006. 1
[4] Gray J, Distributed computing economics[J], Queue, 2004, 6(3): 63-68
[5] Armbrust M, Fox A, Griffith R, et al. Above the clouds: A Berkeley view of cloud computing, UCB/EECS-2009-28[R]. Berkeley: EECS Department, 2009
[6] Cisco Visual Networking, Cisco global cloud index: Forecast and methodology 2015-2020, CISCO White paper, 2015
[7] Ha K, Pillai P, Lewis G, et al. The Impact of Mobile Multimedia Applications on Data Center Consolidation[C]// IEEE International Conference on Cloud Engineering. IEEE, 2012:166-176
[8] Hu W, Gao Y, Ha K, et al. Quantifying the Impact of Edge Computing on Mobile Applications[C]// ACM Sigops Asia-Pacific Workshop on Systems. ACM, 2016:5
[9] Satyanarayanan M. The Emergence of Edge Computing[J]. Computer, 2017, 50(1): 30-39
[10] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]//Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012: 13-16
[11] Yi S, Li C, Li Q, A survey of fog computing: Concepts, applications and issues [C]. ACM: The 2015 Workshop on Mobile Big Data, 2015: 37–42
[12] Satyanarayanan M, Bahl P, Caceres R, et al. The Case for VM-Based Cloudlets in Mobile Computing. [J]. IEEE Pervasive Computing, 2009, 8(4):14-23
[13] Ahmed A, Ahmed E, A Survey on Mobile Edge Computing[C]// IEEE International Conference on Intelligent Systems and Control. IEEE, 2016
[14] Hong K, Lillethun D, Ramachandran U, et al. Mobile fog: A programming model for large-scale applications on the internet of things[C]. ACM: The Second ACM SIGCOMM Workshop on Mobile Cloud Computing, 2013: 15–20