A Service Composition Mechanism Based on Mobile Edge Computing for IoT

Danmei Niu[1,2], Yuxiang Li[1,3], Zhiyong Zhang[1,3], Bin Song[1,3]
[1] School of Information Engineering, Henan University of Science and Technology, Luoyang, China
[2]State key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
[3]Henan International Joint Laboratory of Cyberspace Security Applications, Luoyang, China
E-mail: niudanmei@163.com, liyuxiang@haust.edu.cn, xidianzzy@126.com, songbin@haust.edu.cn

Abstract—**In the past few years, different kinds of Internet of Things (IoT) devices have become more intelligent and have more functions. Some services provided by IoT devices have been successfully deployed in actual scenarios. However, the introduction of cloud computing into these services has created some limitations. By providing services close to users, mobile edge computing can support various network applications and services, reduce latency and avoid overload of cloud center nodes. This paper designs a service composition mechanism based on mobile edge computing for IoT. The simulation results show that this composition can effectively reduce service execution time. It provides an efficient and reliable solution for service composition mechanism.**

Keywords—service composition; mobile edge computing; Internet of Things; cloud computing

I. INTRODUCTION

Cloud computing develops rapidly and is applied in various fields [1]. Cloud computing center can provide a channel and medium for data communication among users and Internet of Things (IoT) devices, such as smart phones, tablets, sensors and smart home appliances. IoT devices directly contact with remote cloud data centers for service requests and submissions, which often resulting in high latency and overload of network bandwidth [2]. A large number of IoT devices generate considerable communication traffic. If mobile edge computing is adopted, service requests can be processed before they are sent to the cloud [3, 4].

For example, there is a scenario of service composition based on mobile edge computing in IoT [5]. When a car accident occurs, different roles participate in their own cloud locations, such as traffic light controllers, ambulance managers, fire departments and taxi fleet managers. These participants must be coordinated, well-organized and effectively managed to provide the fastest real-time response.

By providing services close to users, the service requests can be processed before they are sent to the cloud center node [6]. Mobile edge computing can support a variety of network applications and services and reduce communication delays [7]. Mobile edge computing can also avoid overload of cloud center nodes, which can alleviate the shortcomings of cloud computing solutions in IoT [8].

Mobile edge computing is not to replace cloud computing, but to integrate with cloud computing to provide users with better service experience. One of the important purposes of mobile edge computing is to more effectively schedule various service resources, such as computing, storage, communication service resources and so on. Through realizing the scientific management and composition of service resources, we can realize the optimization of equipment utilization, energy consumption, bandwidth, storage et al. Only by reasonably scheduling and allocating cloud, edge server and IoT devices, can the advantages of these three types of resources be brought into full play. Therefore, we can best balance the use of resources at all levels, maximize resource utilization, improve efficiency and satisfy the user's service experience. This paper designs a comprehensive service composition mechanism based on mobile edge computing for IoT.

There are several kinds of studies on service composition. These service composition methods are based on geographical location, user interest similarity or cloud and fog environments.

Through clustering frequently contacted nodes in the same location, the service composition method based on geographical location improves the efficiency of service composition. Girolami et al. [9] put forward a service discovery protocol based on human mobility and social behavior, which improves the efficiency of service selection and reduces the delay of message transmission. Service composition method based on interest similarity is to guide service search through the interest of nodes in the process of service selection, which always find nodes with high interest similarity. Qureshi et al. [10] propose a query method of introducing user interest in mobile ad hoc network environment. There are also some studies on composed service management in cloud and fog environments. Xavi et al. [5] introduce a layered cloud architecture and its advantages, as well as the open issues and research challenges. It also proposes a practical case of collaborative management.

The mechanism of this paper will overcome the shortcomings of limited ability of a single node and the response delay in cloud computing environment. It will also strengthen the effective composition among the service resources of IoT devices, mobile edge servers and cloud center nodes.

II. SYSTEM MODEL

Figure 1 shows the system model, which is divided into three layers, that is, there are three types of entities involved in transmitting composed services to users. These entities include IoT devices such as smart phones, laptops, tablets and various types of sensors, mobile edge devices such as mobile edge server connect to cellular network, cloud center nodes or cloud servers.

978-1-7281-5712-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICISCE48695.2019.00197

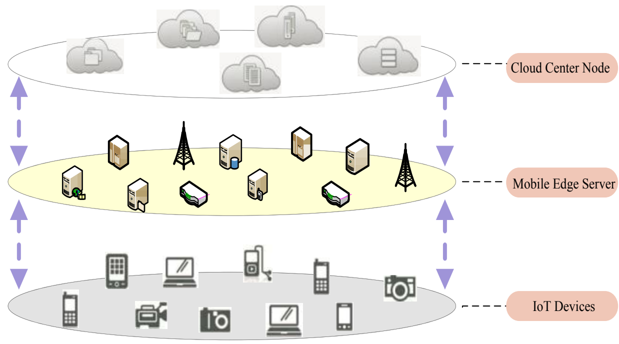

Figure 1. System model

Service composition may require nodes of different layers in various scenarios. As shown in Figure 1, if the service does not require too much processing capability or requires low latency for collected data from real-time sensors, it can be processed directly at the mobile edge server closest to the user. Several mobile edge servers are interconnected in adjacent areas, so that IoT devices in their respective areas can work together to accomplish service composition tasks. This will avoid overload of network and cloud data center and reduce the response time of returning services from cloud center nodes. If the requested service requires a large amount of computing resources or storage resources, and mobile edge server is not competent enough, it needs the service resources provided by the traditional cloud center node. IoT devices can request cloud center node to cooperate through mobile edge server.

III. A SERVICE COMPOSITION MECHANISM BASED ON MOBILE EDGE COMPUTING

Through mobile edge servers or peripheral terminals in IoT, users can get the required data or services closer to themselves, instead of always requesting data or services from cloud center nodes. This way will shorten the service response time and improve the data access ratio and quality of service. Therefore, frequently accessed service resources can be cached on mobile edge servers or IoT devices, and replicas and cached data can be updated periodically through notifications sent by cloud center nodes.

Because there is a large number of service resources residing in the cloud center nodes and many files are large, it is not the best solution for the cloud center nodes to copy all files to the mobile edge servers. Essentially, only highly requested files can be copied to the storage site of mobile edge servers near the users. Large files can be decomposed into smaller files, which can be copied to different mobile edge servers. So users can access data faster according to the location of their IoT devices. Cloud center nodes can also decompose complex service operations into several mobile edge servers. Frequently requested service resources or service operations can be further cached in IoT devices. The smaller service resource or a single service operation can be called atomic service. Both mobile edge servers and IoT devices can notify or update the atomic services they provide. Figure 2 depicts an instance of service composition.

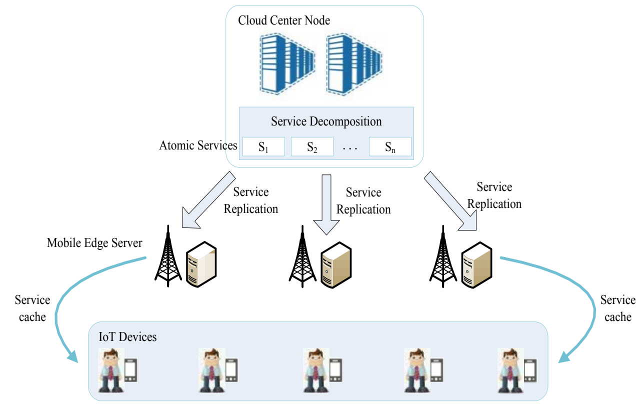

Figure 2. An instance of service composition

The functions of the cloud center nodes on the top level include service decomposition and service replication management. With the help of service decomposition and replication management module, the cloud center node can decompose the frequently accessed services which will be replicated to the mobile edge servers. Service decomposition includes service data decomposition and service operation decomposition. Service replication management module includes service resource discovery module and service replication module. The service resource discovery module is used to discover suitable mobile edge servers. The service replication module replicates the decomposed services in these mobile edge servers.

The functions of mobile edge server in the middle layer include cache node selection, service composition, service notification and service update. Cache node selection module is responsible for discovering the IoT devices suitable for caching services within the communication range of mobile edge servers. It tries to ensure that IoT users can access various services at a closer distance. In other words, IoT users can get services from their adjacent IoT devices or their nearest mobile edge servers. The service request and response time will be de reduced.

According to the memory, residual electricity, computing ability and some other attribute parameters of IoT devices, the cache node selection module of mobile edge server can select better cache nodes in its communication range. The selected cache nodes can be used for service composition. In order to provide users with higher quality composite services, service composition module needs to compose the nodes that can provide atomic services. According to users' service requests, service composition module chooses the required nodes and establishes service paths. Through the notifications sent from the cloud center node, the services replicated on each mobile edge server can be updated periodically.

The functions of the underlying IoT devices include service notification and service update, which can notify the name, type, identification number and some attributes of the service provided by the device. IoT device can also notify its memory, residual electricity, computing power and other performance parameters.

When a user submits the service request to the nearest mobile edge server, the mobile edge server will examine the parameters of the user's service request, such as attribute value description and acceptable quality of service (QoS). Then the mobile edge server decides whether it has enough service resources to satisfy the user's request. If it satisfies the request, it will directly provide services to users. If it does not satisfy the requirements, it will request the mobile edge server or cache node nearby. If there are no needed atomic services, the mobile edge server will transmit the service request to cloud center nodes.

In order to provide collaborative services for users more quickly, the IoT devices register resources with the nearest mobile edge server. According to the resources registered by IoT devices, the mobile edge server can select some devices with better performance and cache some frequently-used service resources or service operations. Therefore, the user's service request will get a faster response. In other words, mobile edge servers need to determine which IoT devices are suitable for caching based on memory, residual electricity, computing ability and transmission capacity. The selected cache nodes provide composed services for users with mobile edge servers and cloud center nodes.

IV. DESIGN OF SERVICE COMPOSITION ALGORITHM

In this paper, the service composition algorithm is as follows.

Step 1: an IoT user sends a service request to his/her nearest mobile edge server.

Step 2: the mobile edge server needs to decompose user's service request to check whether the needed atomic services are available locally.

Step 3: if these atomic services are available locally, the requested atomic services are composed and sent back to the service requester, and the service request succeeds.

Step 4: if there is no atomic service or only part of the atomic services in the mobile edge server, the mobile edge server will communicate with surrounding servers and cache nodes to compose the requested atomic services.

Step 5: if the required atomic services cannot be discovered, the service request will be sent to the cloud center nodes.

Step 6: if there is no required service in the cloud center nodes or the service execution time exceeds the user's requirement, the service request fails.

V. SIMULATION ANALYSIS

The simulation environment and parameters are as follows. This paper adopts Network Simulator Version 3 (NS-3) as a simulation tool. In the simulation area of 2500×1600 (m^2), suppose there is 1 cloud center node which is connected with 8 mobile edge servers. Each mobile edge server connects with 20 IoT devices. Assuming that all the nodes are distributed randomly and follow the random way mobility model (RWP) [11], and the maximum transmission radius of the nodes is 200 m. Suppose that the maximum connection bandwidth of cloud center node is 20 Gbps, mobile edge server is 2 Gbps, and IoT device is 200 Mbps. A service requester can require 4-6 atomic services once. The simulation time is 600s.

A service composition mechanism based on mobile edge server and cache node (MMESCN) presented in this paper is compared with the other two mechanisms. One mechanism is based on mobile edge servers (MMES), which has mobile edge servers, but mobile edge servers only provide services for the IoT devices within their communication range. There is no collaboration between mobile edge servers. The other mechanism is based on cloud center nodes (MCCU), which only provides services for users by cloud center nodes, and no mobile edge servers provides services. The simulation results of the three mechanisms are compared and analyzed from service execution time.

The simulation experiments record the time from each service request sent by a requester to the result received. The service execution time is the sum of the time to accomplish all the service requests. The experiments record all the requests for each service composition. Figure 3 shows the service execution time of three mechanisms MMESCN, MMES and MCCU as the total number of requested atomic services changes. From Figure 3, we can see that the service execution time of the three mechanisms increases with the increase of the number of requested atomic services. When the number of requested atomic services is small, the service provider can accomplish the service requests quickly, and the service execution time is short. As the number of requested atomic services increases, the service execution time of each mechanism increases gradually. Figure 3 shows that when the number of requested atomic services is small, the service execution time of the three mechanisms does not differ significantly. With the increase of the number of requested atomic services, the service execution time of MMESCN increases slower than that of MMES and MCCU.

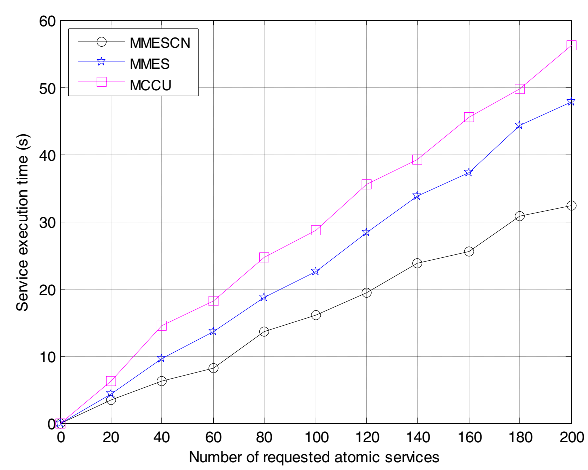

Figure 3. Service execution time changing with number of requested atomic services

Suppose that in the other case, IoT devices are randomly distributed and the number is unstable. IoT devices connect with their nearest mobile edge servers according to the distance. All the service requests in one experiment contain 80 atomic services. Figure 4 shows that how the service execution time of three mechanisms MMESCN, MMES and MCCU changes with the change of number of IoT devices.

As can be seen from Figure 4, when the number of IoT devices is relatively small, the service execution time of the three mechanisms is not significantly different. The service execution time of MCCU is the highest. The service execution time of MMES and MMESCN is similar. Figure 4 shows that the service execution time of MCCU and MMES does not change greatly with the increase of the number of IoT devices. The service execution time of MMESCN decreases gradually.

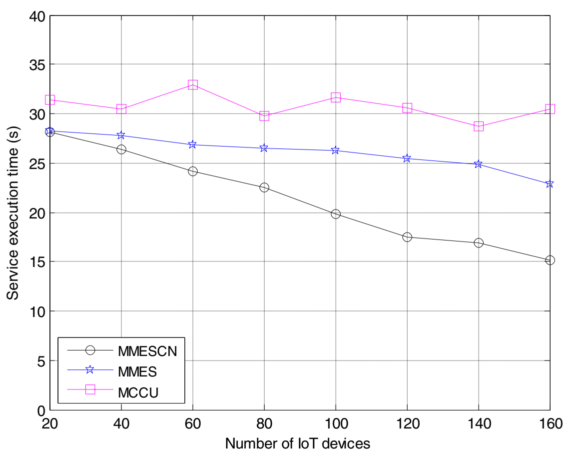

Figure 4. Service execution time changing with number of IoT devices

Compared with the other two mechanisms, MMESCN proposed in this paper can significantly reduce the service execution time with the change of the number of requested atomic services or the number of IoT devices. Therefore, it provides an efficient and reliable solution for service composition mechanism based on mobile edge computing for IoT.

VI. CONCLUSIONS

Mobile edge computing can effectively schedule various service resources, such as computing, storage, communication service resources and so on. Therefore, this paper designs a service composition mechanism based on mobile edge computing for IoT. The simulation experiments show that the method can effectively reduce service execution time. It provides an effective mechanism for service composition based on mobile edge computing.

ACKNOWLEDGMENT

The work was sponsored by National Natural Science Foundation of China Grant No. 61972133 and 61772174, Plan For Scientific Innovation Talent of Henan Province Grant No.174200510011, Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) Grant No. SKLNST-2018-1-09, the Key R&D and Promotion Special Projects (Tackling Hard-nut Problems in Science and Technology) of Henan Provincial Science and Technology Development Plan Grant No. 192102210130 and the Key Scientific Research Projects Plan of Henan Provincial Higher Education Institutions Grant No. 19B520008. We also would like to thank the reviewers and editor for their valuable comments, questions and suggestions.

REFERENCES

[1] Raei H, Yazdani N, Shojaee R. Modeling and performance analysis of cloudlet in Mobile Cloud Computing [J]. Performance Evaluation, 2017, 107: 34-53.

[2] Basu S, Karuppiah M, Selvakumar K, et al. An intelligent/cognitive model of task scheduling for IoT applications [J]. Future Generation Computer Systems. 2018, 88: 254-261.

[3] Niu D M, Rui L L, Huang H Q, et al. A Service Recovery Method based on Trust Evaluation in Mobile Social Network [J]. Multimedia Tools and Applications, 2017, 76 (3): 3255-3277.

[4] Shi W, Dustdar S. The promise of edge computing [J]. Computer，2016, 49 (5): 78-81.

[5] Xavi M B, Eva M T, Ghazal T, et al. Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems [J]. IEEE Wireless Communications, 2016, 23(5): 120-128.

[6] Renkonen K O, Seppala M. Edge analytics in the Internet of things [J]. IEEE Pervasive Computing, 2015, 14 (2): 24-31.

[7] Li W, Santos I, Delicato F C, et al. System modelling and performance evaluation of a three-tier Cloud of Things [J]. Future Generation Computer Systems, 2017, 70: 104-125.

[8] Rimal B P, Van D P, Maier M. Mobile edge computing empowered fiber-wireless access networks in the 5G era [J]. IEEE Communications Magazine, 2017, 55 (2): 192-200.

[9] Girolami M, Barsocchi P, Chessa S, et al. A social-based service discovery protocol for mobile Ad Hoc networks [C]. 12th Annual Mediterranean Ad Hoc Networking Workshop. Ajaccio, 2013: 103-110.

[10] Qureshi B, Min G, Kouvatsos D, et al. An Adaptive Content Sharing Protocol for P2P Mobile Social Networks [C]. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, 2010, 413-418.

[11] Johnson D B, Maltz D A. Dynamic source routing in ad hoc wireless networks [J]. Mobile Computing, 1996, 353: 153-181.