King Saud University

Saudi Journal of Biological Sciences

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

A resource-sharing model based on a repeated game in fog computing

Yan Sun *, Nan Zhang

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

Received 23 October 2016; revised 23 December 2016; accepted 7 January 2017
Available online 27 January 2017

KEYWORDS

Fog computing;
Repeated game;
Crowd-funding algorithm

Abstract With the rapid development of cloud computing techniques, the number of users is undergoing exponential growth. It is difficult for traditional data centers to perform many tasks in real time because of the limited bandwidth of resources. The concept of fog computing is proposed to support traditional cloud computing and to provide cloud services. In fog computing, the resource pool is composed of sporadic distributed resources that are more flexible and movable than a traditional data center. In this paper, we propose a fog computing structure and present a crowd-funding algorithm to integrate spare resources in the network. Furthermore, to encourage more resource owners to share their resources with the resource pool and to supervise the resource supporters as they actively perform their tasks, we propose an incentive mechanism in our algorithm. Simulation results show that our proposed incentive mechanism can effectively reduce the SLA violation rate and accelerate the completion of tasks.

1. Introduction

Cloud computing is a new service mode that can provide available and convenient network visits (Mell and Tim, 2011). It only took several years to integrate in people's lives. At the far cloud end, data centers keep users from the bottom physical framework through virtualization technology and form a virtual resource pool for external services. The cloud data center is composed of many large servers that meet pay-as-you-go demand. These large-scale data centers are constructed by well-capitalized big companies, such as Google, Yahoo, etc. They possess the absolute right of control over resources, and users can only use resources. With the development of mobile internet, more and more heterogeneous devices are connected to the network (Zhang et al., 2011). Although large-scale cloud data centers can meet the complicated requests of users, bandwidth limits may cause network congestion and even service interruptions when many users request services from the data center at the same time. The QoS (quality of service) cannot be ensured if the request has to be processed by the far cloud end. Under this circumstance, fog computing was developed (Bonomi et al., 2012).

Fog computing is a new resource provision mode in which the users not only can use the virtualized resources but can also provide services. In fog computing, some simple requests with high time sensitivity could be processed by geographically

* Corresponding author.
E-mail address: 1590sy@sina.com (Y. Sun).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.sjbs.2017.01.043
1319-562X © 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

distributed devices that can absorb some pressure of the cloud data center. All devices with spare resources can be resource supporters of fog computing, even some sensors and smart phones. Since the resource supporter is closer to the resource consumer, fog computing is superior to cloud computing in terms of response speed.

The resource supporters are all rational and would like to achieve some benefit for their resource contributions. If there is not an effective incentive mechanism, the resource owners will not contribute their resources (Vaquero and Rodero-Merino, 2014). Based on the above problems, the main contributions of this paper are presented as follows:

(1) A system structure based on the neural network of the human body is put forward according to the characteristics of cloud and fog data centers. The reasonability of this system structure is analyzed.
(2) Based on the idea of crowd-funding, a reward and punishment mechanism was established by integrating the computing capacity of geographically distributed devices. This mechanism encourages resource owners to contribute their spare resources and monitors the resource supporters to execute tasks positively; it then increases the working efficiency and reduces the SLA violation rate.

In Section 2, we present the architecture of fog computing based on the human neural network, and describe related issues about crowd-funding. Then, we elaborate the crowd-funding algorithm flow and analyze it mathematically using repeated game theory. In Section 3, simulations are used to show the effects of this algorithm on reducing the SLA violation rate and decreasing task execution time. Our work is concluded, and future research directions are proposed in Section 4. In Section 5, the related work of fog computing is introduced.

2. Related works

Due to continuous development of the internet of things technology, more intelligent devices are used in people's daily lives. These geographically distributed devices possess tremendous idle resources. There are plenty of resources available for users in data centers. Therefore, coordinated management of these resources in a fog environment for automatic deployment, dynamic expansion and distribution according to user needs is a research hotspot.

Many experts and scholars have explored coordinated resources management in the cloud and fog environment. Zhen et al. (2013) introduced the concept of "skewness." By minimizing skewness, the overall utilization of server resources is improved to enhance the ability of the cloud data centers to provide resources to serve the users. They also developed a set of heuristics that effectively prevent system overload and conserve energy. Beloglazov et al. (2012) investigated the issue of virtual machine consolidation in heterogeneous data centers and presented an energy-efficient virtual machine deployment algorithm called MBFD. The algorithm selects the physical machine that increases the energy consumption of the system the least after placing a virtual machine as the destination host where a virtual machine should be placed. The algorithm plays an energy-saving role. Lee and Zomaya, 2012 generated two heuristic algorithms for task integration, ECTCC and MaxUtil. The goal of these heuristic algorithms is to reduce the energy consumption of data centers by improving resource utilization of the physical machines to turn on as few physical machines as possible. Hsu et al. (2014) proposed an energy-aware task consolidation (ETC) technique. The ETC minimizes energy consumption by restricting CPU use below a specified peak threshold and by consolidating tasks among virtual clusters. The network latency when a task migrates to another virtual cluster has been considered in the energy cost model. Gao et al. (2013) investigated the deployment of virtual machines under the homogeneous data center, regarding it as a multi-objective optimization problem. System resource utilization and energy consumption were optimized and a multi-objective ant colony algorithm was presented. Dong et al. (2013) designed a hierarchical heuristic algorithm that considers the communication between virtual machines when analyzing the virtual machine deployment problem. The energy consumption of physical and network resources is optimized. Wu et al. (2014) presented a green energy-efficient scheduling algorithm that efficiently assigns proper resources to users according to the users' requirements in the cloud data center. Their algorithm increases resource utilization by meeting the minimum resource requirement of a job and prevents the excess use of resources. The DVFS technique is used to reduce energy consumption of servers in data centers.

Aazam and Hum (2015a,b) proposed a resources management model based on fog computing. The model in (Aazam and Hum (2015a,b)) considered resource prediction and allocation as well as user type and characteristics in a realistic and dynamic way, thus enabling to adaption to different telecom operators according to requirements. However, their resources management model neglected heterogeneous services, service quality and device movement. In (Aazam and Hum (2015a,b)), the authors proposed a high-efficiency resources management framework. Since fog computing involves different types of objects and devices, how many resources will be consumed and whether request node, device or sensor will make full use of requested resources are unpredictable. Therefore, they developed a resources evaluation and management method by comparing the abandonment probability of fluctuating users to service type and service prices as well as the variance of abandonment probability. This method was conducive to determining correct resource demand and avoiding resource waste. Nevertheless, their resources management model analyzed from the perspective of only the service supplier, and neglected the economic benefits of service users. In (Do et al., 2015), the authors studied resource co-allocation in fog computing and reducing carbon emissions. A high-efficiency distributed algorithm based on the near-end algorithm was developed that decomposed large-scale global problems into several sub problems that can be solved quickly. However, this algorithm only focused on a single data center and neglected the fact that there are multiple small data centers in fog computing. SU et al. analyzed how to share or cache resources between servers effectively using the Steiner tree theory (Su et al., 2015). When the fog server is caching resources, a Steiner tree is produced first to minimize the total path cost. Next, the Steiner tree is compared with a traditional shortest path scheme, which proved that the Steiner tree is more efficient. However, they only analyzed resources management issues between servers and did not perform collaborative analyses on distributed user resources in the fog environment. In (Zeng et al., 2016), the authors designed a high-efficiency task scheduling and resources management strategy designed to minimize time to accomplish tasks in the fog environment to enhance user experiences. For this reason, the authors discussed three problems: (1) how to balance loads between user devices and the computing server, or task scheduling; (2) how to place task images on the storage server, or resources management; and (3) how to balance I/O interrupt requests between storage servers. They were abstracted into a mixed integer nonlinear programing problem. However, the authors basically applied concentrated resources management under cloud computing and did not consider the distributed structural characteristics of fog computing. In (Lee et al., 2016), the authors put forward a gateway conceptual model based on a fog computing framework. This framework mainly consisted of host nodes and slave nodes, managing virtual gateways and resources. This model was suspended in the theoretical study. How to limit found resources in actual application scenarios and determine which resources need virtualization and how to integrate virtual resources have to be solved in the future. In (Song et al., 2016), the authors established a load equilibrium algorithm based on dynamic graph division that could allocate system resources effectively and reduce loss caused by node transferring. This algorithm sacrificed system performance for resource management, which influenced the user experience. In (Wang et al., 2016), the authors introduced the concept of multimedia perception of service and put forward a new resource allocation framework at the cloud edge, or fog end. This framework analyzed the dependence of data in the space, time and frequency domains as well as energy efficiency under different resource allocation strategies considering the effect of a flexible information channel coding rate. Multimedia perception-oriented users designed a physical resource allocation strategy. Their study emphasized data analysis, but did not have a specific resource collaborative management scheme. In (Zeng et al., 2016; Guo et al., 2016), the authors considered an embedded system that was defined by fog computing support software. To enable users to accomplish tasks in minimum time, an efficient resource management strategy was designed. However, this strategy had the disadvantage of overly high computing complexity and poor resource management. To solve the computing complexity problem, in (Gu et al., 2016; Liu, 2013), the authors put forward a two-stage heuristic algorithm based on linear programming that was proven to be highly cost-efficient by experimental results. Nevertheless, most existing research is based on a fixed resources supply model, resulting in low resource flexibility.

The performance of the resources management system is the key to fog computing technology. Devices connected to the network and user demands increase with the continuous development of fog computing, causing resource bottlenecks at the data center. The cloud data center has difficulty meeting the demands of users with high real-time requests. Therefore, it becomes more important to discuss collaborative resources management of data center and network edge devices. Such collaborative management is more complicated because user resources are often distributed and the same resource is often shared by numerous computational nodes. User resources management not only involves topology, configuration, capacity and other intrinsic properties of the network but is also closely related with computing resources, storage resources and distribution of applications. Therefore, studying collaborative management of data center and user resources is challenging and urgent.

3. Methods

For the basic structure of fog computing, most existing research is a three-tiered architecture where fog cloud computing lies between the cloud computing layer and the Internet of Things layer. The fog computing layer is composed of some small data centers, located at the edge of the network where they are closer to users. They can handle relatively simple and high real-time task requirements. We are inspired by Ning and Wang (2011) who proposed a future architecture of the Internet of Things that is similar to a human neural network. The architecture is shown in Fig. 1 and consists of the brain nerve center (cloud data centers), spinal nerve center (fog computing data centers), and the peripheral nerves (smart devices), widely distributed all over the body. The activities of the spinal cord are controlled by the brain.

Peripheral nerves are distributed in the body. They feel stimulation and transfer tasks. The spinal nervous system handles the simple unconditioned reflex, such as the knee jerk reflex. If all requests had to be dealt with by the brain, the brain would be extremely tired. Similar to the characteristics of neural structures of the body, we designed a new system architecture. In our architecture, the intelligent devices can be seen as the peripheral nerves that are widespread geographically, such as the phones, tablets, smart watches, or sensors. The Fog computing center will address some simple and time-sensitive requests (such as the spinal cord knee jerk reflex) that can share the resource pressure of the cloud data center.

The spinal cord is the connecting pathway between peripheral nerves and the brain, which is similar to the location of the fog data center that is the bridge of the underlying Internet of Things and high-level cloud data centers.

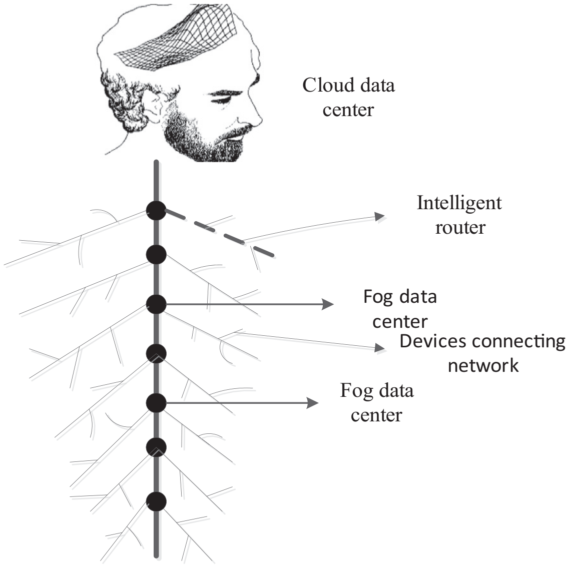

Figure 1 Architecture of fog computing based on the nervous system.

3.1. Game model description

In the open and sharing mobile Internet era, many spare resources are underutilized. In fog computing, users will not take the initiative to contribute their spare resources if there is not an effective incentive mechanism. We established a set of incentive mechanisms based on the idea of crowd-funding and repeated games, and some definitions are as follows:

3.1.1. Definition 1: broker

The local fog computing data centers constructed by small enterprises or universities that have the ability to provide services for users. However, the computing and storage services that they provide are limited. They are eager to improve by integrating the resources of resource supporters.

3.1.2. Definition 2: resources supporters

The resource owners who are willing to contribute some or all of their spare resources and execute tasks assigned by fog data centers are the resource supporters. They can earn rewards by contributing their resource capacities.

3.1.3. Definition 3: crowd-funding reward

α is the financial reward that resource supporters get from the fog broker per unit time by contributing resources.

3.1.4. Definition 4: task reward

β is the financial reward that resource supporters get from the fog broker by performing tasks.

3.1.5. Definition 5: discount factor

δ reflects the degree of patience of players in the game.

3.1.6. Definition 6: self-loss

The self loss ϕ indicates the energy costs and risk costs of crowd-funding supporters when they actively execute tasks. The resource utilization of crowd-funding supporters will improve if they fully use their resources to actively perform a task. System utilization and power consumption are linear according to (Fan et al., 2007; Kusic et al., 2009), which indicates that the increase in system utilization leads to improvement in energy costs. Even if a supporter has some spare resources at present, these resources may be used at another time. Contributing resources will increase the risk of resource shortages on their own devices.

Our crowd-funding algorithm is designed as shown in Fig. 2.

To encourage the resource owners to contribute their resources, the fog broker promises to give the supporters a higher bandwidth if they contribute their spare resources. The additional revenue brought by the higher bandwidth is the crowd-funding reward denoted by α. With the objective of obtaining a higher bandwidth, users will select contributing resources to form a local crowd-funding resource pool. However, after crowd-funding supporters have achieved the benefits, they may refuse to continually provide the resources. To monitor user consistency in contributing resources, we design an incentive mechanism based on the repeated game theory.

First, supporters consider whether to accept the task assigned by the fog broker. If the supporter accepts tasks, they will get a higher reward β. If the user refuses, he can only get α because he is contributing resources ($\alpha < \beta$). If the supporter accepts the task, he will perform the task positively or negatively. The supporters will bring some self-loss ϕ if they perform tasks positively. Suppose the fog broker is unable to know whether the supporter is active and only knows the result of the task. When the task is executed successfully, the income of the fog broker is θ ($\theta > 0$). If the task failed to finish, the income of the fog broker is 0. Suppose when the supporter actively performs the task, the task will surely succeed. When the supporter passively performs the task, the probability of completing the task successfully will be P, and the probability of failure is $1 - P$.

If a task was performed in only one stage, rational supporters will choose performing tasks negatively. Given this, a task will be divided into many stages, so the supporters do not know the end time of the task. Thus, the selection process of resource supporters is equal to an infinite repeated game. To ensure that the supporters perform tasks actively, the supporters will be put on a black list if they do not complete the task in time. This means they will no longer get a reward from the fog broker. Therefore, supporters will make full use of their own resources in order to get more rewards. Then, we design a reasonable trigger strategy according to the concept of the repeated game to motivate and supervise supporters to actively complete tasks. The resources pool of the fog broker has been effectively expanded. It increases capacity for the task and alleviates the pressure on the bandwidth of the cloud data center. Supporters also gain a reward.

3.2. Game analysis for our algorithm

Next, we use repeated game theory to analyze whether the algorithm can effectively motivate supporters to contribute their spare resources and actively perform tasks.

In the incentive mechanism we designed, the game between fog broker and crowd-funding supporters is considered as a repeated game with complete information. Assume that the game has perfect memories meaning game players can remember information about themselves and others. At a random stage G, the player will determine his own strategies based on the strategies of the other side. Here, we first introduce the stage game where a task only executes one stage G.

3.2.1. Stage **G**

The fog broker strategy combination is $\{\beta|\beta \geqslant 0\}$ that gives crowd-funding supporters a reward. The crowd-funding supporter strategy combination is a function from $\{\beta|\beta \geqslant 0\}$ to {actively perform tasks, passively perform tasks, and refuse to perform tasks} that is an infinite strategy dynamic game. In this strategy, actively performing tasks, passively performing tasks or refusing to perform tasks is the response of crowd-funding supporters to how much reward the fog broker has given. Suppose the broker and supporters are rational individuals who are eager to maximize their own benefits. Since the award is paid in advance and contributing their own resources continuously will bring additional costs of energy and risk, if there is not a punitive measure, crowd-funding supporters must select passively performing tasks, and the expected revenue of the fog broker is ($p\theta - \beta$). We assume that $p\theta - \beta < 0$ (performing tasks negatively is difficult to complete the task

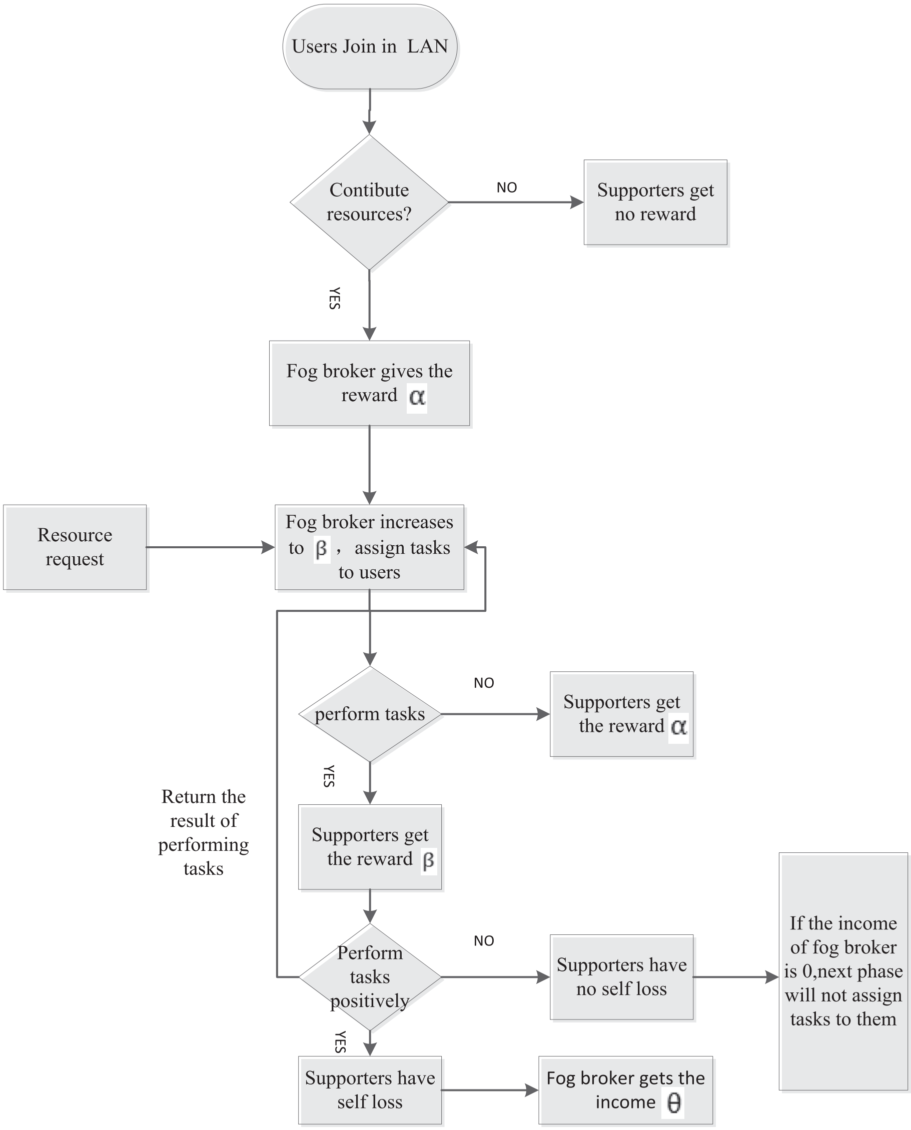
Users Join in LAN
Contibute resources?
NO
Supporters get no reward
YES
Fog broker gives the reward α
Resource request
Fog broker increases to β , assign tasks to users
perform tasks
NO
Supporters get the reward α
YES
Return the result of performing tasks
Supporters get the reward β
Perform tasks positively
NO
Supporters have no self loss
If the income of fog broker is 0,next phase will not assign tasks to them
YES
Supporters have self loss
Fog broker gets the income θ

Figure 2 The flow of the crowd-funding algorithm.

within the required time, therefore, generally p is relatively low). Such being the case, the fog broker will not give any reward to supporters, i.e., $\beta = 0$. Therefore, crowd-funding supporters will certainly select performing tasks negatively. Thus, the Nash equilibrium of stage G is: $\{\beta = 0$, selecting performing tasks negatively when $\beta = 0\}$.

When the model becomes a super game with stage repeats, players can decide their strategies according to the memory of stage G. To avoid fog brokers giving cheap rewards and crowd-funding supporters performing tasks negatively, we design a trigger strategy that is a credible threat to both the fog broker and crowd-funding supporters so that getting rid of this unfavorable situation and reaching Pareto is an excellent outcome.

3.2.2. Trigger strategy T:

- *On the Fog Broker side:* pay a higher reward β^* at the first stage; if the payoff of the fog broker is always θ in the former (t-1) phase, then continue to pay β^*; otherwise no longer give any reward, i.e., $\beta^* = 0$.
- *On the Crowd-funding Supporter side:* if the reward is higher than α, accept tasks assigned by the broker. If the first (t-1) stage rewards are always β^*, then users continue to actively perform tasks at phase t, otherwise execute tasks passively.

The ultimate goal of supporters and the fog broker is to get the highest capital return. Since the supporter does not know at which stage the task ends, equivalently there is an infinite repeated game with no final stage. To ensure the credibility of the trigger strategy, the trigger strategy T needs to satisfy the sub-game refining Nash equilibrium. We will analyze it as follows:

If the players do not deviate from the trigger strategy, the fog data center gives a higher reward β^*, and supporters complete tasks actively, the payoff function of the fog broker in the whole repeated games is:

$$f_w = \theta - \beta^* + \delta(\theta - \beta^*) + \delta^2(\theta - \beta^*) + \dots \tag{1}$$

When resource supporters perform tasks positively, the payoff function is:

$$u_w = \beta^* - \phi + \delta(\beta^* - \phi) + \delta^2(\beta^* - \phi) + \\ = \beta^* - \phi + \frac{\delta}{1-\delta}(\beta^* - \phi) \tag{2}$$

If any player deviates from the trigger strategy, the Nash equilibrium will return. Because the fog datacenter does not give any reward, it will get nothing in return. The payoff function of the fog data center is: $f_z = 0$.

If supporters choose performing tasks negatively, they may complete the task with a weak probability p. Once beyond the longest completion time the requestor can tolerate, the crowd-funding supporter can get no incentive from the fog datacenter, they can only get the basic contribution reward α from the fog broker. The payoff function when supporters perform tasks negatively is:

$$u_z = \beta^* + p\delta u_{z+1} + (1-p)(\delta\alpha + \delta^2\alpha + \cdots) \\ \cong \beta^* + p\delta u_z + (1-p)(\delta\alpha + \delta^2\alpha + \cdots) \tag{3}$$

If the trigger strategy is useful to encourage supporters and brokers, the payoff function u_w of supporters for actively performing tasks should be more than u_z of users for performing tasks negatively. In addition, the payoff function f_w of brokers for giving a high reward should be more than f_z of brokers for giving a low reward.

$$\begin{cases} f_w > f_z \\ u_w > u_z \end{cases} \Rightarrow \begin{cases} \frac{\theta-\beta^*}{1-\delta} > 0 \\ \frac{\beta^*-\phi}{1-\delta} > \frac{\beta^*+\delta(1-p)\frac{\alpha}{1-\delta}}{1-p\delta} \end{cases} \tag{4}$$

(4) can be solved as follows:

$$\alpha + \phi + \frac{1-\delta}{\delta(1-p)}\phi < \beta^* < \theta \tag{5}$$

Therefore, if condition (1.5) is met, the trigger strategy is the Nash equilibrium of the original game. The beginning sub-game between two crowd-funding stages has the same structure with the originally repeated game, which is an infinitely repeated game. Therefore, the triggering strategy is also the Nash equilibrium in the sub-game under condition (1.5). If the stages before the beginning sub-game are all in Nash equilibrium, then the fog broker will still give a high reward β^*. Therefore, the optimal strategy of crowd-funding supporters is to perform tasks positively. Then, supporters must complete the sub-task successfully because of active performance. In subsequent phases, the fog broker will continue to give a high β^* and continue the trigger strategy. The trigger strategy combination in the sub-game is also a Nash equilibrium.

Therefore, this strategy combination is a sub-game perfect Nash equilibrium, which indicates that the trigger mechanism is credible. This strategy combination encourages the game players to maintain cooperation with the fog broker.

4. Results

A small crowd-funding platform was established based on the basic framework of the extended distributed system Hadoop that consisted of a fog broker, a cloud data center and crowd-funding supporters. The crowd-funding supporters were 50 smartphones, which formed a virtual resource pool for external services. The configuration parameters of fog broker and supporters are shown in Tables 1 and 2, respectively. Our simulation mainly detected the SLA violation rate and time for completion of a task under different task loads. Application pressure test data were generated by JMeter. Specific parameter configuration is introduced in the following Tables 1 and 2.

Table 1 Configuration parameters of the fog broker.

CPU	Intel Core 2 DuoE5200
Memory space	4G
Operating system	Win7

Table 2 Configuration parameters of crowd-funding supporters.

CPU	Exynos 8890
RAM	2G
ROM	32G
Operating system	Android 4.4.3
Storage space	16G

The SLA violation rate is defined as the proportion of the number of failed tasks to the number of total tasks. "Failed tasks" means that the supporter failed to complete tasks in the time required by the task requester. As a typical dynamic scheduling algorithm, Min-Min algorithm (Braun et al., 2001) chooses resources by calculating the minimum for double and schedules abundant tasks onto corresponding virtual machines the most quickly, thus enabling completion of all tasks in the shortest time. The MBFD algorithm proposed in (Beloglazov et al., 2012) allocates tasks based on CPU utilization rate. Our resource crowd-funding algorithm encourages resource supporters to contribute their spare resources and process task requests while achieving a bonus in return.

SLA violation rates and the time of completing tasks under different task numbers are shown in Figs. 3 and 4. It can be seen from Fig. 3 that the SLA violation rate with our algorithm is always lower than MM and MBFD. This is because the bonus incentive encourages crowd-funding supporters to make use of idle resources to execute tasks positively and accomplish user task requests in the stipulated time. It will not suffer resource shortages, thus reducing the SLA violation rate. With the increase of tasks, the SLA violation rate of the proposed algorithm showed better stability than the other two algorithms.

The times to accomplish tasks using the three algorithms under different loads are presented in Fig. 4. The proposed algorithm achieved significantly higher execution efficiency than the other two algorithms. This is because crowd-funding users are at the network edges and beyond the restriction of bandwidth, and tasks do not need to be transmitted to the cloud end. With the increase of loads, the proposed algorithm still took less time to accomplish tasks than MM and MBFD.

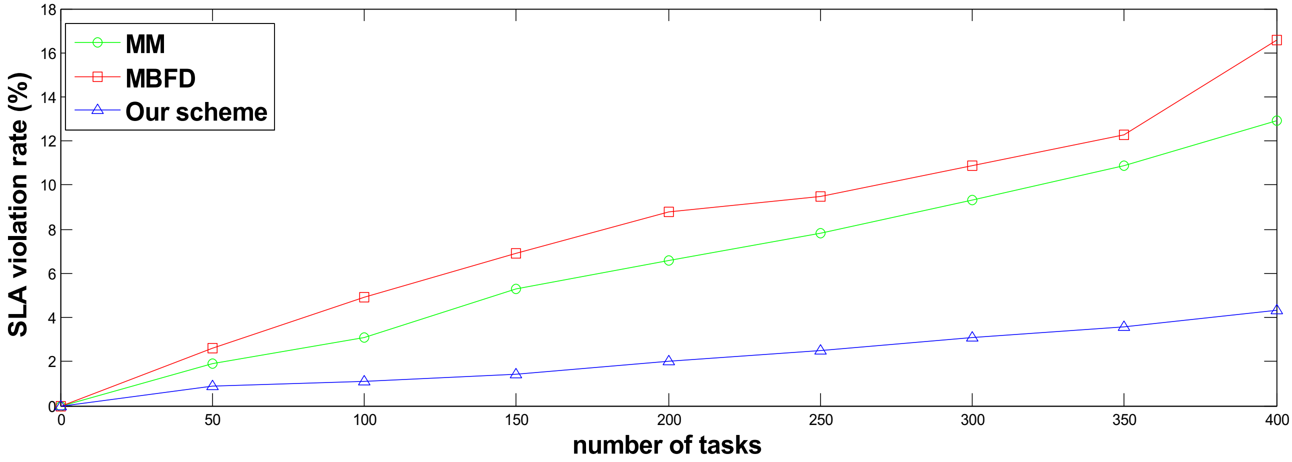

Figure 3 The comparison of SLA violation rates for three schemes with different numbers of tasks.

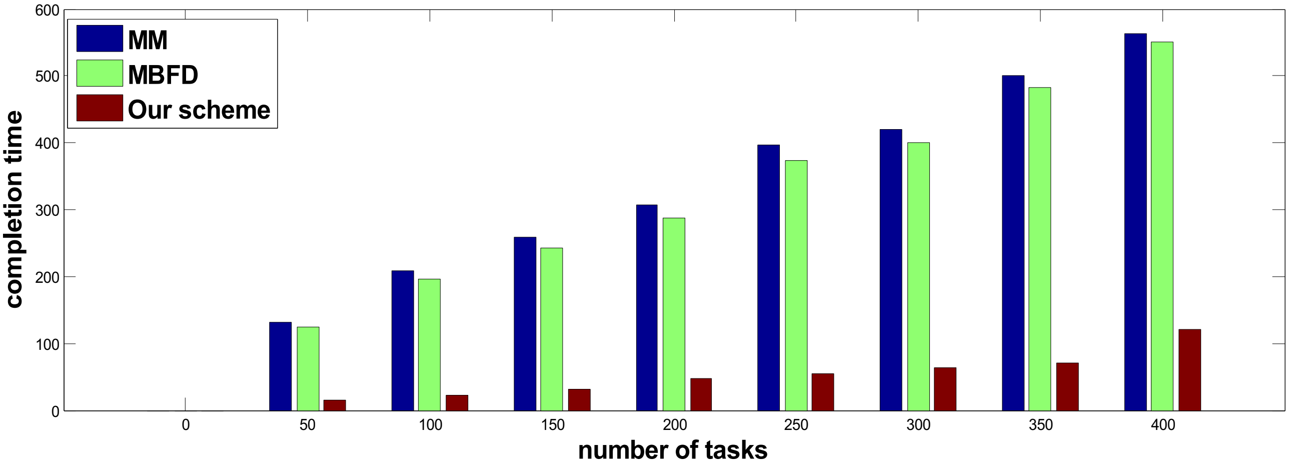

Figure 4 The comparison of completion times for three schemes with different numbers of tasks.

5. Discussion and conclusions

In this paper, we present a system structure based on the neural network of the human body according to the characteristics of cloud and fog data centers. Then, we design a resource crowd-funding algorithm to integrate sporadic resources to form a dynamic resource pool that can optimize the spare resources in the local network. A comprehensive reward and punishment mechanism is presented for the resource supporters in the resource pool. The simulation results shows that our scheme can effectively increase working efficiency and reduce the SLA violation rate by encouraging resource owners to contribute their spare resources and by monitoring the resource supporters to ensure they execute tasks positively. Through this research, we find that unless these widespread devices can work together to create meaningful services, all the resources from devices may be meaningless. Therefore, the integration must be conducted seamlessly and intelligently.

Energy consumption is also an important issue in fog computing systems, and it will be studied in the future. Reducing energy consumption to reduce the costs of service providers is of great significance. Improving the resource utilization of data centers and reducing energy consumption in fog computing data centers will be an important future research direction.

References

Aazam, M., Huh, E. N., 2015. Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, IEEE, pp. 687–694.

Aazam, M., Huh, E. N., 2015. Dynamic resource provisioning through Fog micro datacenter. In: Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on IEEE, pp. 105–110.

Beloglazov, A., Abawajy, J., Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. 28 (5), 755–768.

Bonomi, F., Milito, R., Zhu, J., Addepalli, S., 2012. Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, pp. 13–16.

Braun, T.D., Siegel, H.J., Beck, N., Bölöni, L.L., Maheswaran, M., Reuther, A.I., Freund, R.F., 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. J. Parallel Distrib. Comput. 61 (6), 810–837.

Dong, J., Jin, X., Wang, H., Li, Y., Zhang, P., Cheng, S., 2013. Energy-saving virtual machine placement in cloud data centers. In: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on IEEE, pp. 618–624.

Do, C.T., Tran, N.H., Pham, C., Alam, M.G.R., Son, J.H., Hong, C. S., 2015. A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing. In: 2015 International Conference on Information Networking (ICOIN), IEEE, 324–329.

Fan, X., Weber, W.D., Barroso, L.A., 2007. Power provisioning for a warehouse-sized computer. In: ACM SIGARCH Computer Architecture News, 35(2). ACM, pp. 13–23.

Gao, Y., Guan, H., Qi, Z., Hou, Y., Liu, L., 2013. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. J. Comput. Syst. Sci. 79 (8), 1230–1242.

Gu, L., Zeng, D., Guo, S., Barnawi, A., Xiang, Y., 2016. Cost-Efficient Resource Management in Fog Computing Supported Medical CPS. (99), 1–1.

Guo, H.L., Mu, X.H., Zhao, X.D., Du, F.P., Lenzion, T.V., 2016. The rideability simulation analysis of triangular track conversion system based on multi-body dynamic modeling. J. Mech. Eng. Res. Dev. 39 (2), 492–499.

Hsu, C.H., Slagter, K.D., Chen, S.C., Chung, Y.C., 2014. Optimizing energy consumption with task consolidation in clouds. Inf. Sci. 258, 452–462.

Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G., 2009. Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. 12 (1), 1–15.

Lee, W., Nam, K., Roh, H. G., Kim, S. H., 2016. A gateway based fog computing architecture for wireless sensors and actuator networks. In: 2016 18th International Conference on Advanced Communication Technology (ICACT), IEEE, pp. 210–213.

Lee, Y.C., Zomaya, A.Y., 2012. Energy efficient utilization of resources in cloud computing systems. J. Supercomput. 60 (2), 268–280.

Liu, Z.L., 2013. Fractal theory and application in city size distribution". Inf. Technol. J. 12 (17), 4158–4162.

Mell, P., Tim, G., 2011. The NIST definition of cloud computing.

Ning, H., Wang, Z., 2011. Future internet of things architecture: like mankind neural system or social organization framework? IEEE Commun. Lett. 15 (4), 461–463.

Song, N.N.S., Chao, G., Xingshuo, A., Qiang, Z., 2016. Fog computing dynamic load balancing mechanism based on graph repartitioning. China Commun. 13 (3), 156–164.

Su, J., Lin, F., Zhou, X., Lu, X., 2015. Steiner tree based optimal resource caching scheme in fog computing. China Commun. 12 (8), 161–168.

Vaquero, L.M., Rodero-Merino, L., 2014. Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Comput. Commun. Rev. 44 (5), 27–32.

Wang, W., Wang, Q., Sohraby, K., 2016. Multimedia sensing as a service (MSaaS): exploring resource saving potentials of at cloud-edge IoTs and Fogs. IEEE Internet Things J. 99. 1-1.

Wu, C.M., Chang, R.S., Chan, H.Y., 2014. A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. Fut. Gener. Comput. Syst. 37, 141–147.

Zeng, D., Gu, L., Guo, S., Cheng, Z., Yu, S., 2016. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans. Comput. 99. 1-1.

Zhang, J., Simplot-Ryl, D., Bisdikian, C., Mouftah, H.T., 2011. The internet of things. IEEE Commun. Mag. 49 (11), 30–31.

Zhen, X., Song, W.J., Chen, Q., 2013. Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans. Parallel Distrib. Syst. 24 (6), 1107–1117.