Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2015, Article ID 398709, 9 pages
http://dx.doi.org/10.1155/2015/398709

Research Article

QoE Guarantee Scheme Based on Cooperative Cognitive Cloud and Opportunistic Weight Particle Swarm

Weihang Shi

College of Software Technology, Zhengzhou University, Zhengzhou, Henan 450002, China

Correspondence should be addressed to Weihang Shi; shiwh@zzu.edu.cn

Received 15 July 2015; Revised 8 September 2015; Accepted 14 September 2015

Academic Editor: James Nightingale

It is well known that the Internet application of cloud services may be affected by the inefficiency of cloud computing and inaccurate evaluation of quality of experience (QoE) seriously. In our paper, based on construction algorithms of cooperative cognitive cloud platform and optimization algorithm of opportunities weight particle swarm clustering, the QoE guarantee mechanism was proposed. The mechanism, through the sending users of requests and the cognitive neighbor users' cooperation, combined the cooperation of subcloud platforms and constructed the optimal cloud platform with the different service. At the same time, the particle swarm optimization algorithm could be enhanced dynamically according to all kinds of opportunity request weight, which could optimize the cooperative cognitive cloud platform. Finally, the QoE guarantee scheme was proposed with the opportunity weight particle swarm optimization algorithm and collaborative cognitive cloud platform. The experimental results show that the proposed mechanism compared is superior to the QoE guarantee scheme based on cooperative cloud and QoE guarantee scheme based on particle swarm optimization, compared with optimization fitness and high cloud computing service execution efficiency and high throughput performance advantages.

1. Introduction

With the rapid development of cloud computing technology and diversification of user requirements of mobile Internet, how to provide the scalable service and how to optimize the hardware and software platforms have been the hot research issue [1]. Particularly, according to the computing service of the cloud platform [2], we have to obtain the comprehensive understanding of user experience requirements and quality evaluation, which could not only create the maximum interests of services providers but also satisfy the requirements of users. The cloud platform could be adjusted with the real-time dynamic state information adaptively, to further enhance the cloud computing service support capabilities. A series of research results have been obtained, such as Seamless QoE Support [3], Context-Aware QoE [4], and Policy-Based and QoE-Aware Content Delivery [5].

About cooperative cognition technology and cloud computing algorithm, Kaewpuang et al. [6] provided the guarantee of mobile applications by studying and sharing the radio and computing with mobile cloud computing environment. Lei et al. [7] proposed a novel cognitive cooperative vehicular ad hoc network for solving the contradiction between the increasing demand of diverse vehicular wireless applications and the shortage of spectrum resource. Feteiha and Hassanein [8] addressed the area of heterogeneous wireless relaying vehicular clouds and devised an advanced vehicular relaying technique for enhanced connectivity in densely populated urban areas. The spectrum leasing strategy based on cooperative relaying for cognitive radio networks was proposed in [9]. The implementation of the CCRN framework applied to IEEE 802.11 WLANs was proposed by [10].

On the other hand, the fast cloud-based web service composition approach was proposed according to the characteristics of notion of Skyline [11]. The task-based system load balancing method using particle swarm optimization (TBSLB-PSO) was proposed for achieving system load balancing by only transferring extra tasks from an overloaded VM instead of migrating the entire overloaded VM [12].

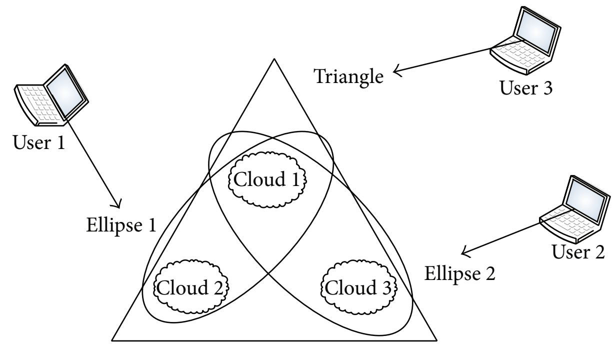


Figure 1: Architecture of cloud platform.

This paper puts forward a cloud theory-based particle swarm optimization (CTPSO) algorithm proposed by Ma and Xu [13], which is used to solve a variant of the vehicle routing problem. Multiple Strategies Based Orthogonal Design PSO was presented by Qin et al. [14], which is used with a small probability to construct a new exemplar in each iteration. The hybrid methods for fuzzy clustering were proposed in [15].

In view of the deficiency of the above research results, we studied the cloud platform construction method based on cooperative cognition of data processing units. Then we researched the optimization algorithm with opportunistic weight particle swarm. Finally, we proposed a reliable and efficient QoE guarantee scheme.

The rest of the paper is organized as follows. Section 2 describes the cooperative cognitive cloud. In Section 3, we design the opportunistic weight particle swarm. The QoE guarantee scheme is proposed in Section 4. Experiment results are given in Section 5. Finally, we conclude the paper in Section 6.

2. Cooperative Cognitive Cloud Platform

The cloud platform can store the large scale data. For achieving load balance and dynamic adjustment, the cloud platform would provide different services for different Internet users. The cloud platform can avoid the system performance degradation, which may be caused by user service competition. The cloud platform can also satisfy the service quality reliability simultaneously. The cloud platform architecture is shown in Figure 1.

In Figure 1, there are three clouds in the cloud platform. Every cloud is made up of the computer. According to the needs of different users, the sub platform is different. The service of User 1 is provided by elliptical 1 cloud platform consisting of cloud 1 and cloud 2. The service of User 2 is provided by elliptical 2 cloud platform consisting of cloud 1 and cloud 3. The service of User 3 is provided by triangle cloud platform consisting of cloud 1, cloud 2, and cloud 3.

Cloud platform is defined as the state vector $S_{\text{cp}} = \{C_N, \text{RAM}_{\text{MAX}}, \text{RAM}_{\text{AVE}}, \text{DIS}_{\text{MAX}}, \text{DIS}_{\text{AVE}}, \text{CH}_{\text{MAX}}, \text{CH}_{\text{AVE}}, \text{CPU}_{\text{MAX}}, \text{CPU}_{\text{AVE}}\}$.

Here, C_N denotes the number of computers in subcloud platform, RAM_{MAX} denotes the total member size of subcloud platform, RAM_{AVE} denotes the average member size, DIS_{MAX} denotes the total disk size, DIS_{AVE} denotes the average disk size, CH_{MAX} denotes the maximum channel bandwidth, CH_{AVE} denotes the average channel bandwidth, CPU_{MAX} denotes the maximum CPU operating frequency, and CPU_{AVE} denotes the average CPU operating frequency.

The user cloud computing service request is set to 1. The request includes 4 metrics, which are E_{RAM}, E_{DIS}, E_{CH}, and E_{CPU}. Here, $E_{\text{RAM}} + E_{\text{DIS}} + E_{\text{CH}} + E_{\text{CPU}} = 1$.

The matching vector F between cloud platform and user demand is defined as $\{C_N, E_{\text{RAM}} * \text{RAM}_{\text{MAX}}, (E_{\text{RAM}}/C_N) * \text{RAM}_{\text{AVE}}, E_{\text{DIS}} * \text{DIS}_{\text{MAX}}, (E_{\text{DIS}}/C_N) * \text{DIS}_{\text{AVE}}, E_{\text{CH}} * \text{CH}_{\text{MAX}}, (E_{\text{CH}}/C_N)\text{CH}_{\text{AVE}}, E_{\text{CPU}} * \text{CPU}_{\text{MAX}}, (E_{\text{CPU}}/C_N)\text{CPU}_{\text{AVE}}\}$.

Hence, cloud platform and user request evaluation results are as shown in the following formula:

$$\begin{aligned} a &= E_{\text{RAM}}\text{RAM}_{\text{MAX}} + E_{\text{DIS}}\text{DIS}_{\text{MAX}}, \\ b &= E_{\text{CH}}\text{CH}_{\text{MAX}} + E_{\text{CPU}}\text{CPU}_{\text{MAX}}, \\ E_{(S_{\text{cp}},\text{User})} &= \frac{a+b}{C_N}. \end{aligned} \tag{1}$$

In order to satisfy the request of users, the cloud platform load is shown in the following:

$$\begin{aligned} \text{RAM}_T &= \sum_{t=1}^{T}\sum_{i=1}^{C_N} E^t{}_{\text{RAM}}\text{RAM}^t_{\text{AVE}}, \\ \text{DIS}_T &= \sum_{t=1}^{T}\sum_{i=1}^{C_N} E^t{}_{\text{DIS}}\text{DIS}^t_{\text{AVE}}, \\ \text{CH}_T &= \sum_{t=1}^{T}\sum_{i=1}^{C_N} E^t{}_{\text{CH}}\text{RAM}^t_{\text{CH}}, \\ \text{CPU}_T &= \sum_{t=1}^{T}\sum_{i=1}^{C_N} E^t{}_{\text{CPU}}\text{RAM}^t_{\text{CPU}}. \end{aligned} \tag{2}$$

Whether the cloud platform is able to satisfy the user request or not could be judged by the following formula:

$$\begin{aligned} &Y \ \left(\mathrm{RAM}_T + \mathrm{DIS}_T + \mathrm{CH}_T + \mathrm{CPU}_T\right) \le E_{(S_{\mathrm{CP}},\mathrm{User})}, \\ &N \ \left(\mathrm{RAM}_T + \mathrm{DIS}_T + \mathrm{CH}_T + \mathrm{CPU}_T\right) > E_{(S_{\mathrm{CP}},\mathrm{User})}. \end{aligned} \tag{3}$$

Here, Y denotes that the could platform can satisfy the requirements and N denotes that it cannot do this.

After the subcloud platform satisfies the requirements, the user should broadcast the signal to clouds. The neighbor users could listen to the channel and receive the signal. The collection of service requests for multiple neighbor users is shown in the following formula:

$$U_R^{N_U} = \bigcup_{i=1}^{N_U} \sum_{i=1}^{4} E_i {U^i}_R. \tag{4}$$

Here, $U_R^{N_U}$ denotes the collection of user cognitive service requests, N_U denote the user number of joining the cognitive networks, E denote the 4 aspects of comprehensive evaluation of user's cloud computing service request, and U_R denote the evaluation value of a neighbor's service request.

The channel bandwidth is shown in formula (5) between the sending request user and the cognitive users:

$$\begin{aligned} S_T &= \sum_{i=1}^{C_N} {\mathrm{CH}^i}_T, \\ S_{\mathrm{AVE}} &= \frac{a+b}{C_N}, \\ a &= E_{\mathrm{RAM}} \mathrm{RAM}_{\mathrm{AVE}} + E_{\mathrm{DIS}} \mathrm{DIS}_{\mathrm{AVE}}, \\ b &= E_{\mathrm{CH}} \mathrm{CH}_{\mathrm{AVE}} + E_{\mathrm{CPU}} \mathrm{CPU}_{\mathrm{AVE}}, \\ B_W &= \sum_{i=1}^{N_U} \sum_{j=1}^{C_N} \ln\left({S^i}_T {S^j}_{\mathrm{AVE}}\right). \end{aligned} \tag{5}$$

Here, S_T is used to analyze the resource of subcloud platform and S_{AVE} is used to analyze the average data processing ability of subcloud platform. The channel bandwidth could be obtained by the combination of neighbor users and subcloud platform resources. So, the cloud platform could be constructed based on the cooperative cognition of sending request user, cognitive users, and computers of subcloud platform. The architecture is shown in Figure 2.

To sum up, the optimization of cloud resource management platform could be provided by the cooperative control of cloud platform data processing units and cooperative transmission of the sending request user and the cognition neighbor users, which is shown in the following formula:

$$\begin{aligned} &\text{minimize:} \quad \sum_{i=1}^{\mathrm{SCP}_N} {S^i}_{\mathrm{CP}} \\ &\text{subject to:} \quad \begin{cases} \sum_{i=1}^{N_U} \sum_{j=1}^{C_N} \ln\left({E^i}_{\mathrm{RAM}} {\mathrm{RAM}^j}_{\mathrm{MAX}}\right) \ge E\left[\sum_{j=1}^{C_N} {\mathrm{RAM}^j}_{\mathrm{AVE}}\right] \\ \sum_{i=1}^{N_U} \sum_{j=1}^{C_N} \ln\left({E^i}_{\mathrm{DIS}} {\mathrm{DIS}^j}_{\mathrm{MAX}}\right) \ge E\left[\sum_{j=1}^{C_N} {\mathrm{DIS}^j}_{\mathrm{AVE}}\right] \\ \sum_{i=1}^{N_U} \sum_{j=1}^{C_N} \ln\left({E^i}_{\mathrm{CH}} {\mathrm{CH}^j}_{\mathrm{MAX}}\right) \ge E\left[\sum_{j=1}^{C_N} {\mathrm{CH}^j}_{\mathrm{AVE}}\right] \\ \sum_{i=1}^{N_U} \sum_{j=1}^{C_N} \ln\left({E^i}_{\mathrm{CPU}} {\mathrm{CPU}^j}_{\mathrm{MAX}}\right) \ge E\left[\sum_{j=1}^{C_N} {\mathrm{CPU}^j}_{\mathrm{AVE}}\right]. \end{cases} \end{aligned} \tag{6}$$

Here, the function E is used to compute the mean of all data processing unit of the relevant resources in cloud platform.

3. Opportunistic Weight Particle Swarm

Through considering the various types of user requests and the opportunistic weight, we used and improved the particle swarm optimization algorithm to realize the optimization objectives of cooperative cognitive cloud platform and guarantee quality of the user experience.

Assume that the cognitive cooperative cloud platform is a particle swarm and composed of m particles, which is denoted as $\mathrm{CC} = \{\mathrm{CC}_1, \mathrm{CC}_2, \ldots, \mathrm{CC}_m\}$.

The j particle expresses the data service progress of the data packet, which is denoted by vector $KP = \{KP_1, KP_2, \ldots, KP_K\}$. Here, K denotes the user experience quality of the data packet, which includes the real-time performance, reliability, size, number of hops, and distance.

The sending power of the j particle Pj is $\{P_{j1}, P_{j2}, \ldots, P_{jK}\}$. The extremal optimization cloud platform

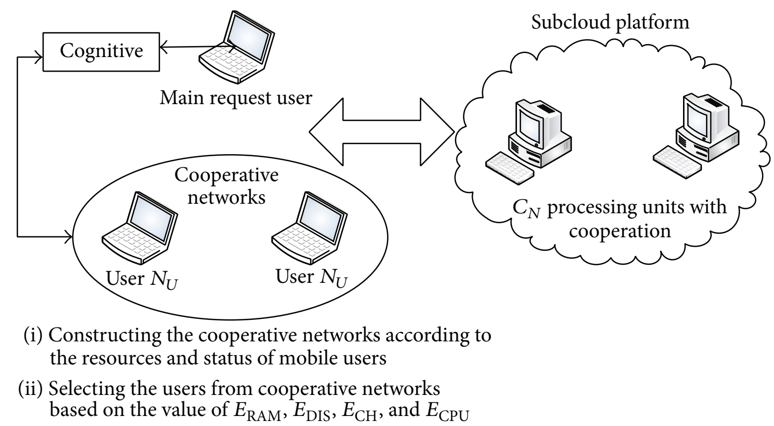


Figure 2: Cooperative cognitive cloud platform.

LO is $\{\text{LO}_1, \text{LO}_2, \ldots, \text{LO}_K\}$. Extremal optimization cloud platform FO is $\{\text{FO}_1, \text{FO}_2, \ldots, \text{FO}_K\}$.

Transmission power opportunity renewal model is shown in the following formula:

$$P_{j+1} = P_j + \sum_{i=1}^{K} \lambda_i KP_i + \sqrt{\left|P_j{}^2 - P_{j-1}{}^2\right|}. \quad (7)$$

Here, the optimization of j particle is realized based on j and $j-1$ particle. The sending power could be updated with the opportunistic dynamical scheme.

Resource opportunity renewal model is shown in the following formula:

$$\begin{aligned} \text{RAM}_{j+1} &= \sum_{i=1}^{C_N} E^j{}_{i,\text{RAM}} \sum_{l=1}^{K} KP_l \text{RAM}^j_{i,\text{AVE}}, \\ \text{DIS}_{j+1} &= \sum_{i=1}^{C_N} E^j{}_{i,\text{DIS}} \sum_{l=1}^{K} KP_l \text{DIS}^j_{i,\text{AVE}}, \\ \text{CH}_{j+1} &= \sum_{i=1}^{C_N} E^j{}_{i,\text{CHS}} \sum_{l=1}^{K} KP_l \text{CH}^j_{i,\text{AVE}}, \\ \text{CPU}_{j+1} &= \sum_{i=1}^{C_N} E^j{}_{i,\text{CPU}} \sum_{l=1}^{K} KP_l \text{CPU}^j_{i,\text{AVE}}. \end{aligned} \quad (8)$$

Request weight updating model is shown in the following formula:

$$E^{j+1}{}_{(S_{\text{CP}},\text{User})} = \frac{\sum_{i=1}^{K} KP_i \left(E^j{}_{(S_{\text{CP}},\text{User})} + E^{j-1}{}_{(S_{\text{CP}},\text{User})}\right)}{\sqrt{N_U C_N \left|E^j{}_{(S_{\text{CP}},\text{User})} - E^{j-1}{}_{(S_{\text{CP}},\text{User})}\right|}}. \quad (9)$$

In order to guarantee and sustain dynamic characteristic and diversity of cloud platform in the particle swarm dynamic evolution process, based on subcloud platform optimization extreme and cloud platform extremal optimization, the real-time evaluation function is established and shown in the following formula:

$$\begin{aligned} \text{RH}\big|_{j \to j+l} &= \frac{\sqrt{j+l}}{l} \sum_{i=1}^{K} KP_i \text{RH}_{j+i}, \\ \text{RH}_j &= \log_2 \left(\sum_{i=1}^{K} KP_i\right)^j \sum_{i=1}^{K} \frac{\text{LO}^j{}_i}{\text{FO}_i}. \end{aligned} \quad (10)$$

Flow of the cooperative cognitive cloud platform with opportunity weight particle swarm optimization algorithm is shown in Figure 3; the optimal fitness is shown in Figure 4. From the results of Figure 4, based on the best adaptation degree and the average fitness, this opportunity weight particle group optimization algorithm can satisfy the demand for data optimization cooperative cognitive cloud platform well. With the growing particle size, the best fitness increases gradually and the data clustering effect remains good, which not only enhance the local search ability but also achieve good global optimization effect.

4. QoE Guarantee Scheme

In view of the cooperative cognitive cloud platform, with the opportunity weight particle swarm optimization algorithm, the QoE guarantee mechanism is established from the perspective of the user. The QoE guarantee problem can be transformed into a multiobjective optimization problem, as shown in the following formula:

$$\begin{aligned} &\text{Min} \quad \text{FO}(kp) \\ &\text{s.t.} \quad kp \in KP. \end{aligned} \quad (11)$$

Here, based on the global optimization, the user experience quality guarantee scheme is established based on multiobjective optimization.

Based on the user experience quality optimization model, we defined the user data service expectations of the target function UE and the expected execution efficiency of EE.

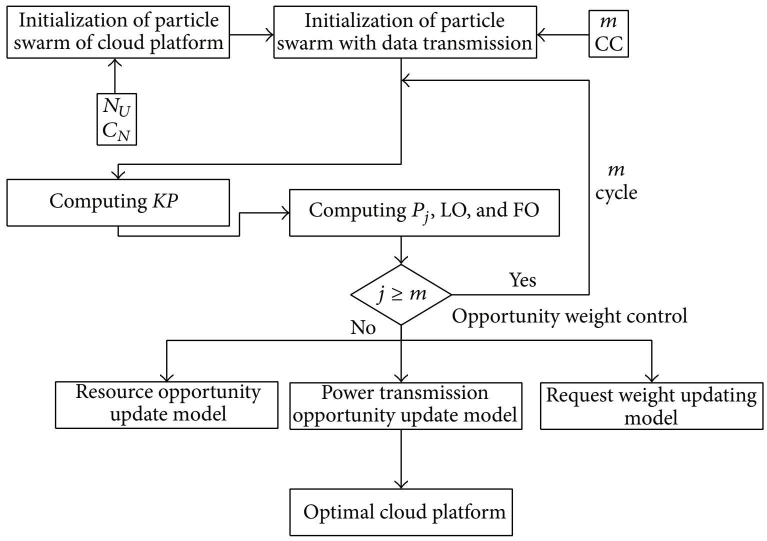


FIGURE 3: Flow of the weight particle swarm optimization algorithm.

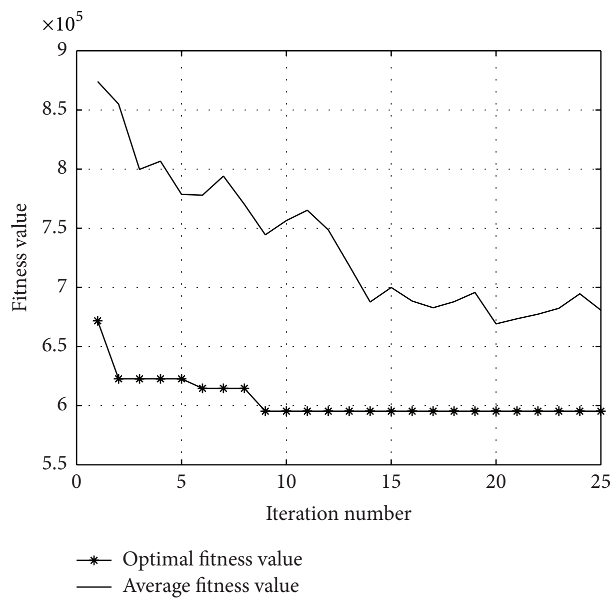


FIGURE 4: Particle swarm optimization.

After sending the service request to cognitive cooperative cloud platform by users, the expectation objective function UE of user data service is used to realize the optimization goal based on the cloud platform during initialization, particle swarm optimization, updating a series of operations, and so forth. This function is shown in formula (12).

The desired execution efficiency of EE: this means the execution time of the data service tasks and the ratio of the required cloud platform resources, which are shown in formula (13):

$$\mathrm{UE} = \ln\left(\frac{\max\left\{\mathrm{RAM}^{j}{}_{\mathrm{MAX}}, \mathrm{DIS}^{j}{}_{\mathrm{MAX}}, \mathrm{CH}^{j}{}_{\mathrm{MAX}}, \mathrm{CPU}^{j}{}_{\mathrm{MAX}}\right\} - S_{\mathrm{AVE}}}{1 + \max\left\{\mathrm{RAM}^{j}{}_{\mathrm{AVE}}, \mathrm{DIS}^{j}{}_{\mathrm{AVE}}, \mathrm{CH}^{j}{}_{\mathrm{AVE}}, \mathrm{CPU}^{j}{}_{\mathrm{AVE}}\right\}}\right), \tag{12}$$

$$\mathrm{EE} = \ln\left(\frac{\mathrm{UE} - \sum_{i=1}^{K} KP_i \left(E^{j}{}_{(S_{\mathrm{CP}},\mathrm{User})} + E^{j-1}{}_{(S_{\mathrm{CP}},\mathrm{User})}\right)}{1 + \left|E^{j}{}_{(S_{\mathrm{CP}},\mathrm{User})} - E^{j-1}{}_{(S_{\mathrm{CP}},\mathrm{User})}\right|}\right). \tag{13}$$

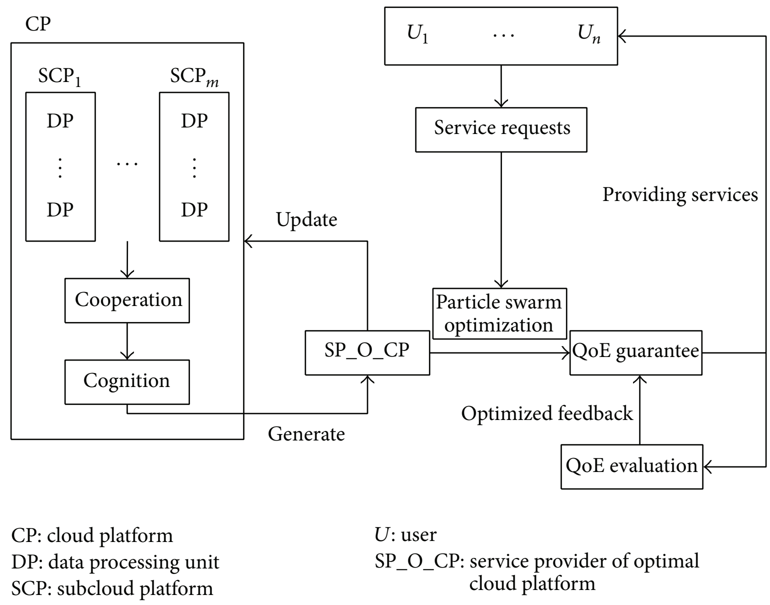


FIGURE 5: QoE guarantee mechanism architecture.

Above all, based on formulas (11), (12), and (13), the objectives of QoE guarantee mechanism are shown in formula (14), which could minimize the parameters of UE, P, and FO and maximize EE:

$$\begin{aligned}
\text{Min UE} &= \ln \sum_{i=1}^{K} KP_i \left(\frac{\max\left\{\text{RAM}^j{}_{\text{MAX}}, \text{DIS}^j{}_{\text{MAX}}, \text{CH}^j{}_{\text{MAX}}, \text{CPU}^j{}_{\text{MAX}}\right\} - S_{\text{AVE}}}{1 + \max\left\{\text{RAM}^j{}_{\text{AVE}}, \text{DIS}^j{}_{\text{AVE}}, \text{CH}^j{}_{\text{AVE}}, \text{CPU}^j{}_{\text{AVE}}\right\}} \right), \\
\text{Max EE} &= \ln \left(\frac{\text{UE} - \left(E^j{}_{(S_{\text{CP}},\text{User})} + E^{j-1}{}_{(S_{\text{CP}},\text{User})} \right)}{1 + \left| E^j{}_{(S_{\text{CP}},\text{User})} - E^{j-1}{}_{(S_{\text{CP}},\text{User})} \right|} \right)^{1/\sum_{i=1}^{K} KP_i}, \\
\text{Min } P &= \frac{1}{P_N} \sum_{j=1}^{P_N} \lambda_j \sum_{i=1}^{K} KP_i + \sqrt{\left| P_j{}^2 - P_{j-1}{}^2 \right|} P_j, \\
\text{Min FO} &= \frac{\alpha \text{UE} + \beta \text{EE} + \gamma P}{\sqrt{\alpha^2 \beta^2 \gamma^2}}.
\end{aligned} \tag{14}$$

The architecture is shown in Figure 5. The workflow of the proposed scheme is described as follows.

Step 1. Construct the cooperative cognitive cloud platform according to the cloud platform state and user requests.

Step 2. Divide the subcloud platform by selecting cooperative users and updating the cloud state.

Step 3. Execute the opportunistic weight particle swarm algorithm.

Step 4. Establish the QoE guarantee mechanism from the user's point of view.

5. Performance Evaluation

In order to validate the performance of the QoE guarantee mechanism based on cooperative cognitive clouds and opportunity weight particle swarm (Q-CCC-OWPW), we designed and developed 6 clouds in the rectangular area of 4000 meters ∗ 4500 meters. Each cloud includes several computers. The setting of the experiment is shown in Table 1. The topology of experiment is illustrated as in Figure 6.

The QoE guarantee scheme based on opportunity weight particle swarm (Q-OWPW) and QoE guarantee scheme based on cooperative cognitive clouds (Q-CCC) and Q-CCC-OWPW are compared and analyzed from the aspects of

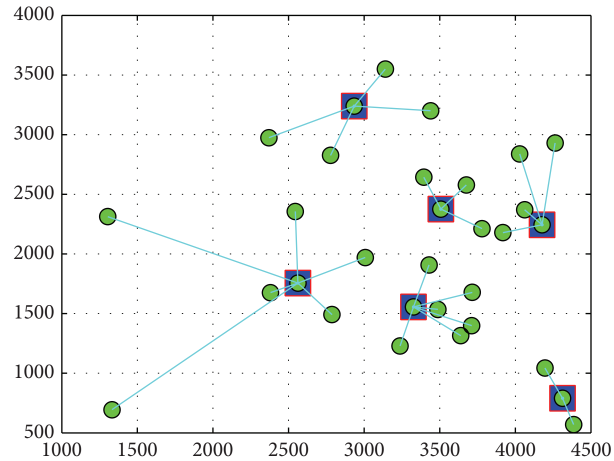


Figure 6: Experiment platform.

Table 1: Cloud settings.

Cloud	Number of computers	Location	Configuration	Disk usage
1	4	(2500, 2400)	Same	30%
2	5	(3400, 2600)	Same	10%
3	5	(4100, 2400)	Same	20%
4	3	(2400, 1700)	Same	25%
5	6	(3500, 1600)	Same	15%
6	3	(4300, 800)	Same	35%

optimal adaptive degree, throughput rate, and user's demand error and execution efficiency. The results are shown in Figure 7.

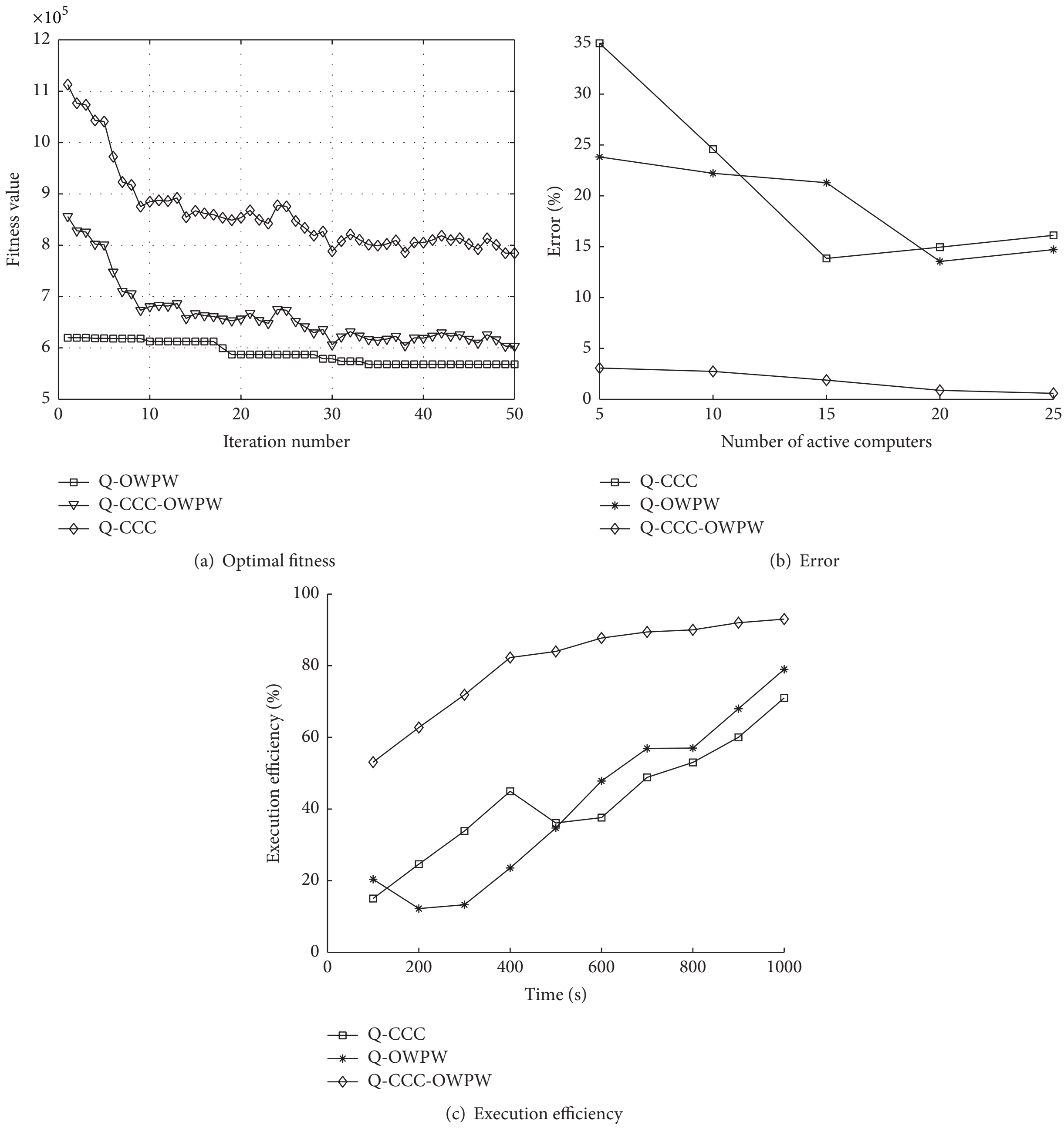
With the increase of the number of iterations, the optimal fitness curves of three QoE guarantee mechanisms are shown in Figure 7(a). Among them, Q-CCC-OWPW has the best fitness and gradually tends to be stable. This has benefited from the cooperation of the data processing computer and the opportunity particle swarm optimization, not only the perception of the cloud platform resources and dynamic adjustment according to the channel quality. However, Q-CCC can only sense the channel quality and Q-OWPW can only perceive the cloud platform resources, so the performance is poor.

Figure 7(b) gives the error of the throughput of the three mechanisms of and the throughput of the user's request. It was found that the error of Q-CCC and Q-OWPW is greater than that of the proposed Q-CCC-OWPW. When the active computer number is between 11 and 19, the performance of Q-OWPW is better than the one of Q-CCC. This is because the Q-OWPW cannot obtain the information of cloud resources.

User service request execution efficiency is shown in Figure 7(c). The execution efficiency of the three mechanisms gradually increases as the experimental time increases. This is because the cloud platform initialization, the cloud platform and user channel competition, and resource allocation stage require a certain period of time; after the completion of the above operations, the execution efficiency of the system increased significantly. However, the efficiency of Q-OWPW prior to 500 s is lower than that of Q-CCC, which is due to the long time required for the optimization of particle swarm optimization. After more than 500 s, the efficiency of Q-CCC is higher than that of Q-OWPW. This is because the particle swarm optimization algorithm obtains the gain of data transmission. For Q-CCC-OWPW, the construction of the opportunity weight particle swarm optimization, to achieve global optimization, has higher efficiency.

6. Conclusions

In order to improve the execution efficiency of cloud and guarantee the quality of experience, we put forward the efficient and reliable QoE guarantee mechanism based on the cooperative cognitive cloud platform construction algorithms and opportunities weight particle swarm clustering optimization algorithm. First, according to the sending requests user and the cognitive neighbor users, the computers of subcloud platform through cooperative cognitive scheme cloud platform would be constructed and updated. Then, in order to realize the optimization objectives of cooperative cognitive cloud platform and improve quality of user experience, particle swarm optimization algorithm was improved for clustering by considering the user end of various types of requests the weight changes in the dynamic of opportunities. Finally, based on the cooperative cognitive cloud platform and the opportunity weight particle swarm optimization algorithm, the QoE guarantee mechanism was proposed from the user's point of view. The experimental results show that the proposed mechanism has obvious advantages in optimizing the degree of adaptation, the efficiency of the implementation of cloud computing services, and the throughput rate.



(a) Optimal fitness

(b) Error

(c) Execution efficiency

Figure 7: QoE guarantee performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2883–2894, 2013.

[2] J. He, Y. Wen, J. Huang, and D. Wu, "On the cost-QoE tradeoff for cloud-based video streaming under Amazon EC2's pricing models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 669–680, 2014.

[3] G. Kim, S. Lee, and S. G. Lee, "Seamless QoE support for mobile cloud services using IEEE802.21 MIH and the GENI future internet framework," *International Journal of Software Engineering and Knowledge Engineering*, vol. 24, no. 7, pp. 1039–1063, 2014.

[4] K. Mitra, A. Zaslavsky, and C. Ahlund, "Context-aware QoE modelling, measurement, and prediction in mobile computing systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 920–936, 2015.

[5] M. Olli, Y. F. Zarrar, M. Petteri, and J. Lessmann, "Policy-based and QoE-aware content delivery using Q-learning method,"

Wireless Personal Communications, vol. 83, no. 1, pp. 315–342, 2015.

[6] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.

[7] Z. Lei, L. Tao, L. Wei, Z. Siting, and L. Jianfeng, "Cooperative spectrum allocation with QoS support in cognitive cooperative vehicular ad hoc networks," *China Communications*, vol. 11, no. 10, pp. 49–59, 2014.

[8] M. F. Feteiha and H. S. Hassanein, "Enabling cooperative relaying VANET clouds over LTE-A networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1468–1479, 2015.

[9] K. Ma, J. Yang, G. Hu, and X. Guan, "Cooperative Relay-Aware Spectrum leasing based on Nash bargaining solution in cognitive radio networks," *International Journal of Communication Systems*, vol. 28, no. 7, pp. 1250–1264, 2015.

[10] M. B. Pandian, M. L. Sichitiu, and H. Dai, "Optimal resource allocation in random access cooperative cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1245–1258, 2015.

[11] S. Wang, Q. Sun, H. Zou, and F. Yang, "Particle swarm optimization with skyline operator for fast cloud-based web service composition," *Mobile Networks & Applications*, vol. 18, no. 1, pp. 116–121, 2013.

[12] F. Ramezani, J. Lu, and F. K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 739–754, 2014.

[13] Y. Ma and J. Xu, "A cloud theory-based particle swarm optimization for multiple decision maker vehicle routing problems with fuzzy random time windows," *Engineering Optimization*, vol. 47, no. 6, pp. 825–842, 2015.

[14] Q. Qin, S. Cheng, Q. Zhang, Y. Wei, and Y. Shi, "Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization," *Computers & Operations Research*, vol. 60, pp. 91–110, 2015.

[15] T. M. Silva Filho, B. A. Pimentel, R. M. C. R. Souza, and A. L. I. Oliveira, "Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6315–6328, 2015.