

Game Based SBCA Cloud Resource Adjustment Method

Yan Gao*, Bin Zhang**, Mengzhu Lu*, Anxiang Ma*

* School of Computer Science and Engineering, Northeastern University, China

** Software College Northeastern University, China

gaoyan@mail.neu.edu.cn, zhangbin@mail.neu.edu.cn, 1326882789@qq.com, maanxiang@mail.neu.edu.cn

Abstract—The performance of SBCA (SOA Based Cloud Applications, a typical form of cloud applications) can be adjusted by changing the occupied amount of resources. How to adjust the amount of resources occupied by SBCA that can meet the SLA constraints with the minimum resource adjustment cost has become the focus of cloud computing research area. In the process of SBCA resource adjustment, SBCA providers and resource providers both want to choose the adjustment plan with maximizing profit for themselves, so there has a conflict of interest between the two providers. Aiming at the problem, this paper proposes a game based SBCA cloud resource adjustment method. The method presents a game based representation model of SBCA cloud resource adjustment problem, gives the utility calculation method of both sides of the game, and solves the problem by using fireworks algorithm. The experiments show the effectiveness of the proposed method.

Keywords—SBCA, Resource adjustment, Game theory, Fireworks algorithm, Cloud computing

I. INTRODUCTION

With the development of cloud computing, more and more software systems are deployed as SaaS to provide services in the cloud environment^[1]. The optimization and management of SaaS layer applications (i.e. cloud applications) has become an important research area in the field of cloud computing^[2]. The cloud application provides services to users by signing a Service Level Agreement (SLA) with users^[1,3]. However, the dynamically changing environments may cause the performance of cloud application to deviate from SLA constraints. The performance of cloud application can be adjusted by increasing or decreasing the amount of resources it occupies. Cloud computing uses a pay-on-demand model, so how to adjust the amount of resources occupied by the cloud application can meet the SLA constraints with the minimum resource adjustment cost has become the focus of cloud computing research area^[4].

In the cloud environment, the software generation model based on SOA (Service-Oriented Architecture) has become an important method of building cloud applications^[5]. SOA-based Cloud Application (SBCA) combines multiple component services according to certain business logic, and its performance is directly affected by the performance of component services. Component services are deployed on virtual machine (VM), and its performance is closely related to the resources of the virtual machine. By increasing or

decreasing the resources amount of the VM, the performance of component services can be adjusted^[3]. Therefore, when the cloud environment changes, the resources amount of VM occupied by the component service can be adaptively adjusted to change its performance, thereby realizing the adjustment of the SBCA performance, that is, the performance of the whole SBCA can be guaranteed by adaptively adjusting the resources occupied by component services to meet the SLA constraints. Therefore, most researchers use the method of changing the amount of VM resources occupied by component services to adjust their performance, so as to achieve the purpose of performance adjustment for the entire SBCA^[3, 6-9].

In the process of SBCA resource adjustment, SBCA providers propose the adjustment scheme to resource providers (i.e. VM providers) with the goal of maximizing profit, but resource providers will also consider the corresponding adjustment costs of the adjustment scheme to make corresponding decisions, and there has a conflict of interest between the two providers. Aiming at the conflict of interest between SBCA providers and cloud resource providers in the process of resource adjustment of SBCA, this paper proposes a game based SBCA cloud resource adjustment method. The method presents a game based representation model of SBCA cloud resource adjustment problem, gives the utility calculation method of both sides of the game, and solves the problem by using fireworks algorithm.

II. SBCA RESOURCE ADJUSTMENT PROBLEM

The VM resource adjustment operation may increase or decrease the amount values of VM resource attributes, for example change RAM from 4M to 8M and storage from 10GB to 15GB. That we call as RAS (Resource Adjustment Schema). A component service may have multiple RAS depending on the resource adjustment range. A RAS has several items and each item is a change operation for one resource attribute. Different RAS has different variation of resource attributes amount and may cause different expense variation, cost variation and response time variation of component service. All this can be described by RAP (Resource Adjustment Plan). Since the response time of SBCA is a common performance indicator in SLA, this paper

takes response time as an example of SBCA performance indicator to study the dynamic adjustment of SBCA cloud resources.

Definition 2.1 Resource Adjustment Plan (RAP). RAP is represented by a five tuple: $rap_i = \langle ras_{ij}, \Delta t, \Delta e, \Delta p, c \rangle$, where rap_i represents RAP of component service s_i ; ras_{ij} represents the j -th resource adjustment schema of s_i ; Δt represents the change variation of s_i response time corresponding to ras_{ij} ; Δe represents the change variation of expense corresponding to ras_{ij} ; Δp represents the change variation of profit corresponding to ras_{ij} ; c represents the resource adjustment cost corresponding to ras_{ij} .

All rap s for a component service made up the Candidate Resource Adjustment Plan Set (CRAPS). In general, SBCA providers always want to get the maximum profit at the least expense. Therefore, resource adjustment of SBCA is to select a rap that has the biggest profit for SBCA from each component service CRAPS, and make up the resource adjustment plan of SBCA. Therefore, the SBCA cloud resource adjustment problem in cloud environment can be modelled as a type of optimization problem, as shown in formula 1.

$$\begin{cases} \text{Max}(\Delta P_{SBCA}) \\ \text{s.t. } \Delta T_{SBCA} < \Delta T^c \end{cases} \quad (1)$$

where, ΔP_{SBCA} represents the profit change variation of SBCA providers; ΔT_{SBCA} represents the change variation of SBCA response time after resource adjustment, positive number means that the response time will be extended after resource adjustment, and negative number means that response time will be shortened after resource adjustment; ΔT^c represents the constraint of SBCA response time variation that the adjustment must reach, and the positive and negative numbers have the same meaning as ΔT_{SBCA} .

ΔT_{SBCA} is aggregated by the response time change variation of the component services in SBCA, as shown in formula 2.

$$\Delta T_{SBCA} = CalT(\Delta t_i) \quad (2)$$

where, $CalT(\Delta t_i)$ is a function to calculate response time of SBCA according to the component services in it, and its calculation method can be referred to [10]; Δt_i represents the change variation of response time of service s_i .

III. MODELING OF SBCA RESOURCE ADJUSTMENT PROBLEM BASED ON GAME THEORY

A. Game based SBCA Resources Adjustment Problem Model

The resource adjustment operation of each component service in the SBCA will be performed by the resource provider. For different ras , the adjustment methods that the resource provider adopts may different, and the adjustment cost may also different. When the SBCA performance changes, according to the SLA constraint, the service provider will propose an adjustment plan to the resource provider with the goal of maximizing profit, but the resource provider will also consider the adjustment cost corresponding to the adjustment plan to make a corresponding decision. Suppose that the SBCA provider sp selects rap_x that maximizes its own profit.

Therefore, if the resource provider rp adjusts resource according to rap_x , it needs to start a new physical machine, and the adjustment cost to be paid is 10, but the corresponding profit is 8. The profit for sp is less than the cost, so rp will not agree to the adjustment plan rap_x .

We can see from the above example that both the SBCA provider and the resource provider will choose a resource adjustment scheme that maximizes their own profit while selecting resource adjustment scheme. So the two parties have conflicts of interest and form a contradiction. In order to solve this problem, this paper proposes to convert the optimal resource adjustment scheme selection problem into game theory problem and establish a game between SBCA provider and resource provider. Thus, the formula 1 is converted to formula 3.

$$\begin{cases} \text{Max}(US_{SBCA}) = \text{Max}(\sum_{i=1}^m \sum_{j=1}^{n_i} us_{ij} * g_{ij} * x_{ij}) \\ \text{Max}(UR_{SBCA}) = \text{Max}(\sum_{i=1}^m \sum_{j=1}^{n_i} ur_{ij} * g_{ij} * x_{ij}) \end{cases} \quad \text{s.t.} \quad (3)$$

$$\Delta T \leq \Delta T^c$$

$$g_{ij} \in \{0,1\}, \quad i = 1, 2, \dots, m;$$

$$\sum_{j=1}^{n_i} x_{ij} = 1, \quad x_i \in \{0,1\}, \quad i = 1, 2, \dots, m;$$

where, Where US_{SBCA} represents the profit obtained by SBCA provider, UR_{SBCA} represents the profit obtained by resource provider, us_{ij} and ur_{ij} represent the profit obtained by the SBCA provider and the profit obtained by the resource provider after game under the j -th rap of service s_i ; g_{ij} represents game result the j -th rap of service s_i , if the game of the j -th rap has a Nash equilibrium solution, then $g_{ij}=1$, otherwise $g_{ij}=0$; m represents the number of service in SBCA; n_i represents the number of rap in CRAPS of service s_i ; x_{ij} represents the choice of the j -th rap in CRAPS of service s_i , $x_{ij}=1$ represents the adjustment choose the j -th rap , and $x_{ij}=0$ represents the adjustment does not choose the j -th rap , and each service can choose only one rap in its own CRAPS.

B. Gamers utility calculation method for SBCA resource adjustment

There are two players in the game model for SBCA resource adjustment: SBCA providers and cloud source providers, which are abbreviated as s and r respectively. The strategy space of the game model consists of all participants' strategy sets. For the problem of SBCA resources adjustment, S_s represents the strategy set of the SBCA provider, and S_r represents the strategy set of the cloud resource provider. Literature [11] proposed a game based method for services selection problem in Service Computing to solve the conflict problem between service provider and service consumer. The method has reference value for solving our problem.

According to a selected rap of a service, the strategy of SBCA provider is to apply for adjusting (*apply*), not to apply for adjusting (*unapply*), and the strategy of resource provider is to adjust (*adjust*) and not to adjust (*unadjust*).

The two gamers will obtain different utility when they choose different strategy. The utility matrix is used to

represent the different utility obtained by two gamers under different strategies. For the j -th *rap* of services S_i , the utility matrix can be expressed as formula 4.

$$SRM_{ij} = \begin{bmatrix} \langle us_{00}, ur_{00} \rangle_{ij} & \langle us_{01}, ur_{01} \rangle_{ij} \\ \langle us_{10}, ur_{10} \rangle_{ij} & \langle us_{11}, ur_{11} \rangle_{ij} \end{bmatrix} \quad (4)$$

where, us_{xy} and represents the utility obtained by the SBCA provider; ur_{xy} represents the utility obtained by the cloud resource provider; x and y have values of 0 and 1, 0 denotes the strategy of *apply* or *adjust*, and 1 denotes *unapply* or an *unadjust*.

For each *rap* of each component service in SBCA, the utility matrix is established and the Nash equilibrium solution is solved. Then, formula 3 is solved according to all *raps* with Nash equilibrium solutions.

For the j -th *rap* of services S_i , the utility of the SBCA provider and the cloud resource provider under different strategies is calculated as follows.

1) The SBCA provider applies to adjust the j -th *rap*, and the cloud resource provider adjusts the j -th *rap*.

$$\langle us_{00}, ur_{00} \rangle_{ij} = \langle \frac{rap_{ij} \cdot \Delta p - rap_{ij} \cdot \Delta e}{rap_{ij} \cdot \Delta e}, \frac{rap_{ij} \cdot \Delta e - rap_{ij} \cdot c}{rap_{ij} \cdot c} \rangle \quad (5)$$

2) The SBCA provider applies to adjust the j -th *rap*, and the cloud resource provider does not adjust the j -th *rap*.

$$\langle us_{01}, ur_{01} \rangle_{ij} = \langle \frac{rap_{ij} \cdot \Delta p - rap_{ij} \cdot \Delta e}{rap_{ij} \cdot \Delta e}, -\eta \frac{rap_{ij} \cdot \Delta e - rap_{ij} \cdot c}{rap_{ij} \cdot c} \rangle \quad (6)$$

3) The SBCA provider does not apply to adjust the j -th *rap*, and the cloud resource provider adjusts the j -th *rap*.

$$\langle us_{10}, ur_{10} \rangle_{ij} = \langle -\eta \frac{rap_{ij} \cdot \Delta p - rap_{ij} \cdot \Delta e}{rap_{ij} \cdot \Delta e}, \frac{rap_{ij} \cdot \Delta e - rap_{ij} \cdot c}{rap_{ij} \cdot c} \rangle \quad (7)$$

4) The SBCA provider does not apply to adjust the j -th *rap*, and the cloud resource provider does not adjust the j -th *rap*.

$$\langle us_{11}, ur_{11} \rangle_{ij} = \langle -\eta \frac{rap_{ij} \cdot \Delta p - rap_{ij} \cdot \Delta e}{rap_{ij} \cdot \Delta e}, -\eta \frac{rap_{ij} \cdot \Delta e - rap_{ij} \cdot c}{rap_{ij} \cdot c} \rangle \quad (8)$$

In the above formula, η represents the penalty factor, and $\eta > 1$, indicating that the rejection will have a large negative impact; $rap_{ij} \cdot \Delta e$ is calculated according to pricing method of cloud resource provider; $rap_{ij} \cdot \Delta p$ is calculated by using formula 9.

$$rap_{ij} \cdot \Delta p = \frac{rap_{ij} \cdot \Delta t}{\Delta T_{SBCA}} * \Delta P_{SBCA} \quad (9)$$

where, ΔP_{SBCA} represents the total profit change variation of SBCA. There are many ways^[12-13] to calculate the profit ΔP_{SBCA} according to the SBCA performance change amount. This paper uses a stepwise function to calculate ΔP_{SBCA} based on the response time of SBCA. The calculation formula is shown in the formula 10.

$$\Delta P_{SBCA} = \begin{cases} P_0, & 0 < \Delta T_{SBCA} \leq \Delta T^c \\ P_0 - k \cdot (\Delta T_{SBCA} - \Delta T^c) / \Delta T^c, & \Delta T^c < \Delta T_{SBCA} \leq \Delta T^{mc} \\ 0, & \Delta T_{SBCA} > \Delta T^{mc} \end{cases} \quad (10)$$

where, P_0 represents the corresponding profit for the response time within the SLA constraint time ΔT^c ; k is a constant, represents the the penalty coefficient; ΔT^{mc} represents the maximum limit of the SBCA response time, if the response time is between ΔT^c and ΔT^{mc} , the profit will be stepped down as the response time increases; if the response time

exceeds ΔT^{mc} , the profit is zero. This paper uses the negative of the expense of SBCA ΔE_{SBCA} to calculate the profit P_0 , as shown in formula 11.

$$P_0 = -\Delta E_{SBCA} = -\sum_{i=1}^m \sum_{j=1}^{n_i} rap_{ij} \cdot \Delta e * x_{ij} \quad (11)$$

According to the different adjustment methods of resource providers, the calculation method of resource adjustment cost c is also different. There are three types of resource adjustment methods for VM providers: resource adjustment on the current physical machine; need to do VM migrate migration, but does not need to start a new physical machine; need to do VM migrate migration, and need to start a new physical machine. These three different resource adjustment methods have a large gap in the adjustment cost. The adjustment cost of j -th *rap* of services s_i calculation method is shown in the formula 12.

$$rap_{ij} \cdot c = \alpha \cdot rap_{ij} \cdot \Delta e \quad (12)$$

where, α represents the cost coefficient, is set differently according to the adjustment operation of resource provider. In this paper, if resource provider only need to adjust the resources on the current physical machine, then $\alpha=1$; if resource provider need to perform virtual machine migration, but do not need to start a new physical machine, then $\alpha=2$; if need to start a new physical machine, then $\alpha=3$.

IV. FA BASED PROBLEM SOLVING METHOD

SBCA is composed of multiple component services according to certain business logic. The business logic includes sequence, parallel, selection and loop, etc., so that the relationship between SBCA performance and component service performance is nonlinear. It is difficult to accurately decompose the SLA constraints of the whole SBCA into each component service. Therefore, the SBCA cloud resource dynamic adjustment problem needs to be solved from the perspective of global optimization. This paper uses the FA (Fireworks Algorithm) to solve the problem. Firework Algorithms^[14] is a swarm intelligence algorithm inspired by the natural phenomenon that fireworks explode in the night sky to produce sparks and illuminate the surrounding area. The algorithm regards spark as a feasible solution in the solution space and the process of generating a certain number of sparks from fireworks explosion as its search domain. The fireworks algorithm has been successfully applied to solve the optimization problem in many fields^[15], and achieved good results.

1) Individual spark coding design

Each spark represents an SBCA resource adjustment plan, and an SBCA resource adjustment plan is composed of *rap* of all component services. Therefore, the number of component services in SBCA is the dimension of the spark. Each dimension has a number to represent *sap* serial number in CRAPS of a component service. And the serial number is generated randomly in range of 1 to the max account of CRAPS for each component service. The spark coding mode is shown in Figure 1.

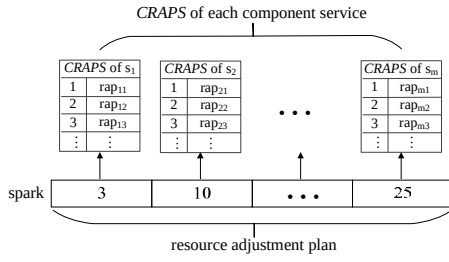


Figure 1. firework coding mode

2) Fitness function design

This paper transforms the multi-objective optimization problem into a single-objective optimization problem. The fitness value of the individual fireworks is calculated by the formula 13.

$$f(x) = \begin{cases} \alpha * US_{SBCA} + \beta * UR_{SBCA}, & \Delta T_{SBCA} \leq \Delta T^c \\ 0, & \Delta T_{SBCA} > \Delta T^c \end{cases} \quad (13)$$

where, α and β are constants, indicating the effect of US_{SBCA} and UR_{SBCA} on the final fitness value. In this paper, both α and β are set to 0.5.

3) Iterative termination condition design

In order to avoid the problem that the search time is too long due to the excessive sparks population, this paper chooses to stop the operation when the number of sparks meets a certain amount. This paper set the max sparks to 80.

V. EXPERIMENTS ANALYSIS

In order to verify the effectiveness of the proposed method, DFWA^[16] (Dynamic Search Fireworks Algorithm) and IFWA^[17] (Improved Fireworks Algorithm) are used to solve the problem respectively, and then the experimental results of two different algorithms are compared. The effects of the number of component services, the number of component service resource adjustment schemes and constraint factor on the algorithm are analyzed respectively.

As the global constraints are strict or not have great influence on the algorithm, this paper uses constraint factor^[18] to measure the global constraints strict extent. The calculation formula of constraint factor for execution time constraint is:

$$\lambda = (\Delta T^c - \Delta T^{\max}) / (\Delta T^{\max} - \Delta T^{\min}) \quad (14)$$

where, λ is constraint factor; $\Delta T^{\max}$ represents the longest response time for SBCA; $\Delta T^{\min}$ represents is the shortest response time for SBCA. $\Delta T^{\max}$ and $\Delta T^{\min}$ are calculated by all rap s of each component services. We can know that $0 \leq \lambda \leq 1$, and when λ is closer to 0, the constraint is more strict^[18].

We use Random Distribution to generate data, including component service CRAPS for each component service. And we use the prediction method we proposed in [19] to generate the response time for each component service RAS, and use liner model to generate cost variation for each component service RAS, thus get the component service CRAPS for each component service.

A. Influence of component service number analysis

We set the number of rap to 10 for each component service and the response time constraint factor to 0.4, the number of

component service is 5, 10, 15, 20, and 25 respectively. The result of DFWA and IFWA are shown in Figure 2.

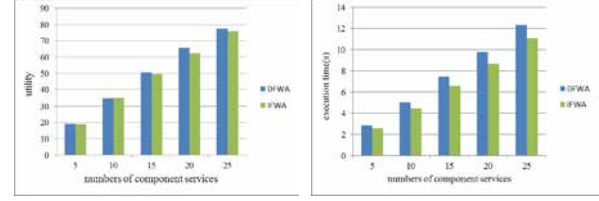


Figure 2. The result of influence of component service number

It can be seen from Figures 2 that when the number of component services is within 15, the optimal utility of the providers searched by the two algorithms is basically the same, and as the number of component services increases, the execution time of each algorithm increases. The DFWA algorithm search works better, but the search time is longer.

B. Influence of component service rap number analysis

The number of component services is 5, the response time constraint factor is 0.4, and the number of rap is 10, 20, 50, 100 and 200 respectively. The result of DFWA and IFWA are shown in Figure 3.

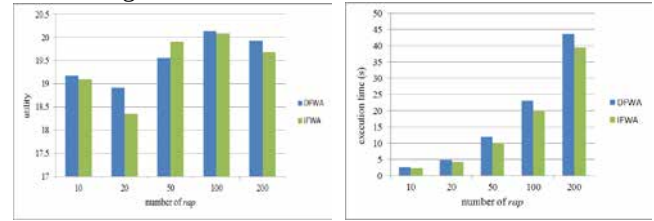


Figure 3. The result of influence of component service rap number

From Figures 3, we can see that with the increase of the number of candidate resource adjustment schemes for component services, the optimal solutions searched by each algorithm are getting better, and the optimal solutions searched by DFWA are better than IFWA, but the execution time is longer.

C. Influence of response time constraint factor analysis

Comparing the optimal solutions of DFWA and IFWA under the execution time constraint factor of 0.2, 0.4, 0.6 and 0.8, respectively. Figures 4 to Figure 6 shows the optimal solutions and the number of component services searched by each algorithm when the number of rap for each component service is 10 and the number of component services is 5, 15 and 25 respectively, and the number of component services is 5, and the number of rap is 10, 100 and 200 respectively.



Figure 4. The result of influence of response time constraint factor (1)

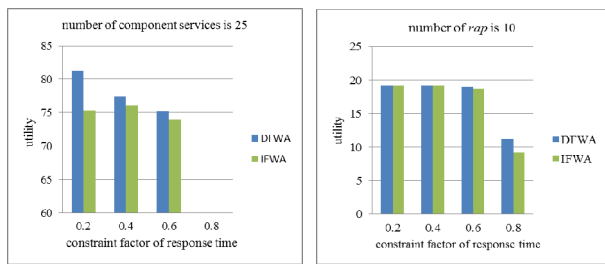


Figure 5. The result of influence of response time constraint factor (2)

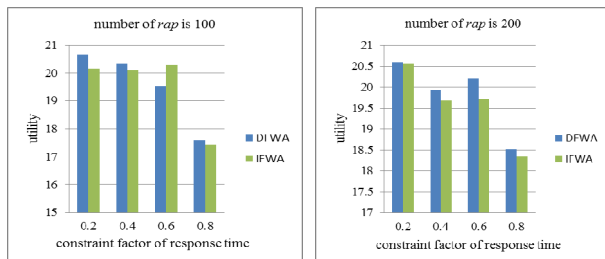


Figure 6. The result of influence of response time constraint factor (3)

Figures 4 to Figure 6 show that when the execution time constraint factor is less than 0.8, the optimal solution searched by each algorithm under different execution time constraint factor does not change much and is relatively stable, but when the execution time constraint factor is greater than 0.8, the search effect of the algorithm is greatly affected, and it is not easy to search for better solution. But in general, the optimal solution of DFWA is better than IFWA.

VI. CONCLUSIONS

How to adjust the SBCA resources amount that can meet the SLA constraints under considering the conflict of interest between SBCA providers and resource providers has become the focus of cloud computing research area. Aiming at the SBCA resources adjustment problem, this paper proposes a game based SBCA cloud resource adjustment method. The method presents a game based representation model of SBCA cloud resource adjustment problem, gives the utility calculation method of both sides of the game, and solves the problem by using two of fireworks algorithms. The experiments show the effectiveness of the proposed method.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation Program of China (61572116, 61572117, 61502089) and by the Fundamental Research Funds for the Central Universities (N171604009).

REFERENCES

- [1] Linlin Wu, Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. SLA-Based Resource Provisioning for Hosted Software-as-a-Service Applications in Cloud Computing Environments[J]. IEEE Transactions on Services Computing, 2014, 7(3): 465-485.
- [2] Alfonso Castro, Víctor A. Villagrà, Beatriz Fuentes, and Begoña Costales. A Flexible Architecture for Service Management in the Cloud[J]. IEEE Transactions on Network and Service Management, 2014, 11(1): 116-125.
- [3] Nallur V, Bahsoon R. A decentralized self-adaptation mechanism for service-based applications in the cloud[J]. Software Engineering, IEEE Transactions on, 2013, 39(5): 591-612.
- [4] Frederico D, Jose C, Anderson F, Vinicius G. A systematic review on cloud computing[J]. The Journal of Supercomputing, 2014, 68(3): 1321-1346.
- [5] Mega C, Waizenegger T, Lebutsch D, Schleipen S, Barney J. Dynamic cloud service topology adaption for minimizing resources while meeting performance goals[J]. IBM Journal of Research and Development, 2014, 58(2): 1-10.
- [6] JiaRao, Yudi Wei, Jiayu Gong, and Cheng-Zhong Xu. QoS Guarantees and Service Differentiation for Dynamic Cloud Applications[J]. IEEE Transactions on Network and Service Management, 2013, 10(1): 43-55.
- [7] Zhicheng Cai, Xiaoping Li, Jatinder N. D. Gupta. Heuristics for provisioning services to workflows in XaaS Clouds[J]. IEEE Transactions on Services Computing, 2016, 9(2): 250-263.
- [8] Ricardo Oda, Daniel Cordeiro, Kelly Rosa Braghetto. Dynamic Resource Provisioning for Scientific Workflow Executions in Clouds[A]. 2018 IEEE International Conference on Services Computing (SCC), 2018: 219-294.
- [9] P. Vinodhialakshmi, R. Anitha. Efficient Dynamic Resource Provisioning using Auction based Technique with Improved QoS in Cloud Environment[A]. 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), 2017: 1-6.
- [10] 36. Jorge Cardoso, Amit P. Sheth. Quality of Service for Workflows and Web Service Processes[J]. Journal of Web Semantics, 2004, 1(3): 281-308.
- [11] Li Ying-qiu, Wen Tao. An Approach of QoS-guaranteed Web Service Composition based on a Win-win Strategy[A]. IEEE 19th International Conference on Web Services (CWS 2012), 2012: 628-630.
- [12] Bhargava H K, Sun D. Pricing under quality of service uncertainty: Market segmentation via statistical QoS guarantees [J]. European Journal of Operational Research, 2008, 191(3): 1189-1204.
- [13] Bhargava H K, Sundaresan S. Contingency Pricing for Information Goods and Services Under Industrywide Performance Standard [J]. Journal of Management Information Systems, 2003, 20(2): 113-136.
- [14] Tan Y, Zhu Y. Fireworks Algorithm for Optimization[A]. International Conference in Swarm Intelligence 2010, 2010: 355-364.
- [15] TAN Ying, Zheng Shaoqiu. Recent advances in fireworks algorithm[J]. CAAI Transactions on Intelligent Systems, 2014, 9(5): 515-528.
- [16] Shaoqiu Zheng, Andreas Janeczek, Junzhi Li, Ying Tan. Dynamic search in fireworks algorithm [A]. 2014 IEEE Congress on Evolutionary Computation (CEC). Beijing, 2014: 3222-3229.
- [17] Liu J, Zheng S, Tan Y. The Improvement on Controlling Exploration and Exploitation of Firework Algorithm [A]. Advances in Swarm Intelligence, 4th International Conference. Harbin, China, 2013: 11-23.
- [18] Gao Jian, Study of QoS Routing Algorithms Based on Immune Mechanism and Genetic Algorithms, Microelectronics & Computer, 2003, 2003(8): 20-21.
- [19] Yan Gao, Si Wang, Bin Zhang, Mengzhu Lu, Xiaopei Wu. A Modelling Method for Mapping the Relationships between Resources and Service Performance Based on I-ABC-ELM[A]. 2016 18th International Conference on Advanced Communication Technology (ICACTION), 2016: 240-243.

Yan Gao received the Ph. D degree in Computer Application Technology from Northeastern University, China, in 2006. Since 2008, he has been an Associate professor in Northeastern University, China. His research area includes Cloud Computing, Service Computing.

Bin Zhang received the Ph. D degree in Computer Science Theory from Northeastern University, China, in 1998. Since 2000, he has been a professor in Northeastern University, China. His research area includes Cloud Computing, Service Computing.

Mengzhu Lu received the master degree in Computer software and Technology from Northeastern University, China, in 2017. Her research area is Cloud Computing.

Anxiang Ma received the Ph. D degree in Computer Application Technology from Northeastern University, China, in 2009. Since 2008, she has been a lecture in Northeastern University, China. Her research area includes Cloud Computing, Service Computing.