Cluster Comput
DOI 10.1007/s10586-017-0907-3

Study of cloud service queuing model based on imbedding Markov chain perspective

ZheXi Yang[1] · **Wei Liu**[1] · **Duo Xu**[1]

Received: 7 February 2017 / Revised: 1 May 2017 / Accepted: 3 May 2017

Abstract A cloud service model has been built in this article to evaluate the QoS of cloud service system. The Markov chain of the viewpoint has been created by applying imbedding Markov chain approach. The random arrivals of cloud requests based on the non follow-up effectiveness characteristics of the Markov chain have been simulated. A cloud service queuing model has been set up by using queue waiting time, network delay time and server processing time as the measurable indicators of cloud service system serving level, and the effectiveness of the evaluating model of cloud service quality raised in this article has been validated by means of analysis and simulation.

Keywords Cloud computing · Response time · Queuing theory · Markov chain

1 Introduction

Cloud computing provides large volumes of integrated hardware and software resources to various users who come randomly, thereby, the study of the QoS (Quality of Service) of cloud computing service systems is similar to the study of the QoS of queuing service (which has indicators like serving time, waiting time, queuing length) [1]. In view of the intense integrated resource utilization and the user usage transparency for the cloud computing, the cloud computing service system is treated as a single server one. But the foundation of cloud computing is distribution and networking, so the cloud computing service system can also be a service system with an infinite number of servers. In this case, the QoS evaluation and the measurement of cloud computing system would be performed in view of one single server (host or virtual host) and its network transmission [2]. Nevertheless, the service level is an abstract indicator which is difficult to quantify, and it is hard to effectively measure the quality of queuing service system by simply using the queuing length; therefore, the customer sustaining time or queuing waiting time in the system is adopted as the indicator to quantify the QoS of the service system explicitly for the purpose of better comparison and analysis [3,20].

✉ ZheXi Yang
13a1903018@cjlu.edu.cn

Wei Liu
11a0904059@cjlu.edu.cn

Duo Xu
xuduo147258@126.com

[1] College of Standardization, China Jiliang University, No. 258, Xueyuan Road, Jianggan District, Hangzhou 310018, China

2 Differences between cloud service queuing systems and classical queuing systems

As a new queuing phenomenon, because of the existing of computer and network carrier and cloud computing center processing methods, cloud service queuing systems have some typical differences compared with classical queuing systems as follows:

2.1 Virtualization

Virtualization is the core technology and main characteristic of cloud computing, so the cloud computing queuing system based on it is definitely virtualized [4]. The access requests distributed around network are received and processed in the cloud service systems uniformly, and the specific software and hardware resources are allocated from the huge hardware resources group of the cloud service systems. So all the

Published online: 19 May 2017

services in the cloud service systems are provided based on the cloud resource virtualization, and the resources acquired are virtual machine related resources, and the server platform accepting access requests are virtual machines [5].

2.2 Multi-source access

The customers of service systems are not typical tangible customers or intangible phone calls, but the web applications or other access requests made by other applications. These kinds of requests may be issued by terminal users, or created by applications, so the sources of access requests coming from the cloud service systems are diversified sources.

2.3 Parallel access

In classical queuing service systems, one server processes one customer access each time, but in cloud computing environment, because of the mechanism of divided time and distributed parallel processing of the computer, one virtual machine can often response to multiple access requests simultaneously, which is not conformed with the service rules of classical queuing theory.

2.4 Peak and valley access

Cloud service systems have been deployed at the cloud end, and users can access it scheduled at $365 \times 7 \times 24$; for specific user groups, there are peak and valley access periods, due to working habits, transaction workflow and time zone differences. The differences of peak and valley time accesses are very important for the design and cost control of cloud service systems [6].

2.5 Co-processing

For improving the efficiency of cloud service, accessing requests often are segmented and processed by task assigning entities, so the single cloud computing request often is handled uniquely in different virtual machines and physical machines, and the complete processing results are gained through co-processing and assembling, etc [7].

2.6 Flexible server configuration

The rapid development of IT leads to the increasing cost reduction, and in turn it causes the host configuration of the cloud service system not only to increase or decrease dynamically by using the corresponding cheap price according to the user requirement, but also to increase and decrease the enabling hosts dynamically supported by the cloud platform which can integrate all software and hardware resources; in the meantime, the maturity of the virtual machine technology makes the requesting and deleting virtual machines dynamically possible. In doing so, the server quantities factor in cloud service system is not the vital issue considered by the designer.

Although it will be a big difference compared with classical queuing service systems, all of them may be transformed into classical queuing service system model by some appropriate adaptations, so cloud service system is still a classical queuing system, the related principles and conclusions of queuing theory are still suitable.

3 Major performance indicators for cloud service queuing systems

Among cloud computing service queuing rules, we take the waiting rules model uniformly, without considering the customer loss. Terminal users issue access requests through Web Applications and other interfaces at the cloud end and wait for the results, which is the waiting queuing process. Based on related studies, the maximum time for users to wait for access results usually is 4–6 s; beyond the limits, the users will leave disappointedly, wihich means the lower cloud service system service level.

From the cloud user perspective, the whole waiting processes, starting from issuing access requests to receiving the process results, are all the waiting time for users to get the responses of the cloud service system. Here, W, the whole waiting time of the cloud customer in the system, represents the service level of the cloud service system. As cloud service system is a super computer, it can response to thousands of access requests, so in the case that there is one data center, we can treat the cloud service system as n parallel and independent server platforms(virtual machines). These platforms build the corresponding service relations with the requests after the tasks have been assigned, or both of them have no any connections [8]. Therefore, the mean value of all waiting time of customers at the service platforms are the whole waiting time of the service systems, and it represents the whole service level of the cloud service system. Let the whole service level of the cloud system be:

$$W = E\left(\frac{1}{n}\sum_{i=1}^{n} W_s^i + T_i\right) \tag{1}$$

In (1) W_s^i denotes the expected stay time for customers staying at the server platforms (it is the sum of waiting time and service processing time), and T_i denotes the network transmission time when customers issue access requests to the server platforms except the stay time at the server platforms.

For the cloud customers, as the access requests are issued, the subsequent tasks are that the cloud service for the provider will be responsible to deliver all the contents, so the delay

time of the information transferring after the access requests have been issued, the queuing waiting time and the requests processing time are all the elapsed time for customers to wait for the responses of cloud service systems [9]. It is the explicit indicator which can be used to measure cloud service quality level for cloud customer, and it is also the important goal for enterprises to refer to and realize when they set up the cloud service. According to the descriptions of formula (1), the parameters of cloud service system service level include stay time and network delay time mainly, and the stay time includes queuing waiting time and service processing time.

4 Waiting time based on Markov chain

Due to the random nature of customers' arrival time and service processing time, the quantity indicators based on the time are all random variables [10]. The transient characteristics of these indicators are not stringency to the queuing system characteristics, so we are usually interested in static quantity indicators in conditions of statistical steady states. When we perform quantity analysis for the performance of queuing system, we analyze the system performances when a system stays in steady states. Considering how cloud users feel when they are in cloud systems, except transmission time consumption, to measure quality of any cloud service systems, we will measure the waiting time for receiving service results after issuing access requests, which is the cloud service system response time.

Because the task information transmission time from cloud resources management system to the virtual machines has been involved in the network delay time in the context, and whatever queuing models we take, it will not affect the value of that time, so the study of queuing waiting time here would study the virtual machine at the end of task responses and processing circumstances of the task queuing waiting processing in front of the queue instead of considering the differences between queuing modes.

Referring to related conclusions of classical queuing theory, for any request arrival queue which does not distinguish its modes, we assume it is a Possion arrival, and its arrival intervals A comply with exponential distribution, and its arrival rate is $1/\lambda$, and service time CDF is $A(x) = prob[A < x]$, pdf is $a(x) = \lambda e^{-\lambda x}$. The task processing time is independent and identically distributed, which usually adopts normal distribution B, and mean value of service is $1/\mu$, and service time CDF is $B(x)=prob[B<x]$, and pdf is $b(x)$. μ can be used to represent the service serving ability of the service syste, and $\rho = \lambda/\mu$ represents the satisfied requests rate per unit time, such as service success rate. W_s still is the mean stay time of requests stay in system, including waiting time and processing time. W_q represents the mean waiting time of queuing users . So, for a single server platform queuing system, we have:

$$W_s = \frac{1}{\mu - \lambda} \quad , \quad W_q = \frac{\rho}{\mu - \lambda} \tag{2}$$

Although from the users perspective, cloud service systems seem like a single server platform queuing service system, for the cloud center with the known huge service capabilities, it actually owns infinite virtual machines to respond to user requests. So, combined with the related theorems of queuing theory, there are mean stay time computing formula for queuing systems with multiple server platforms (the number of server platforms denoted by k) as follows:

$$W_s = \frac{(k\rho)^k \rho}{k!(1-\rho)^2 \lambda} T_0 + \frac{1}{\mu} \tag{3}$$

And the corresponding mean queuing time can be calculated using following formula:

$$W_q = \frac{(k\rho)^k \rho}{k!(1-\rho)^2 \lambda} T_0 \tag{4}$$

Among them

$$T_0 = \left[\sum_{n=0}^{k-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{k!(1-\rho)} \left(\frac{\lambda}{\mu}\right)^k \right]^{-1} \tag{5}$$

Formula (3) is the special form of formula (1), which describes the circumstance that one multi-task platform cloud service system uniformly processes the arrival requests without discrimination.

In this paper, access arrival behaviors and patterns have been appropriately simulated by imbedded Markov chain, instead of Poisson stream model simply. Markov chain can be a common model which is used to describe ordinary discrete event random process frequently [11]. As its memorylessness characteristics, it has been used broadly. The imbedded Markov chain technology involves choosing some Markov points to view the system states, and then, just before the requests arrive, number these task points (including the tasks that are being processed and the tasks that are not), such as 0, 1, 2, …. As such, we will get a homogeneous Markov chain. In this imbedded Markov chain, we obtain the system states before the arrival of each access requests, as is shown in Fig. 1.

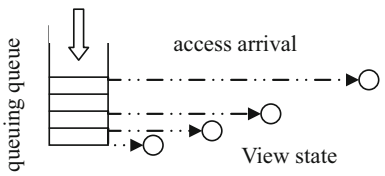


Fig. 1 diagramof imbedded Markov chain view points

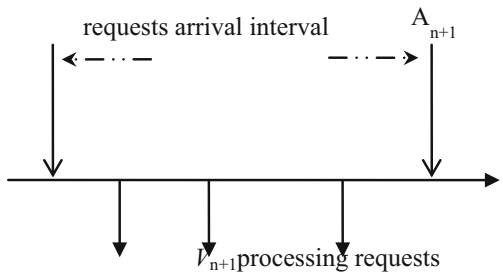


Fig. 2 Diagram of imbedded Madrkov chain process

The arrivals of the access requests still be assumed as complying with Poisson distribution, and each task has a virtual machine responsibel for its response. So the imbedded Markov chain process is shown in Fig. 2.

Among them, A_n and A_{n+1} denote the arrival time that n and $n+1$ access requests reach the system, and q_n and q_{n+1} represent the task numbers corresponding to the arriving time which are being processed and are waiting for being processed in system respectively. Let v_{n+1} represent the numbers of tasks finished from time A_n to A_{n+1}, so we have the following equation:

$$q_{n+1} = q_n - v_{n+1} + 1 \tag{6}$$

The Markov chain transformation probability can be defined as:

$$p_{i,j} = \Pr ob\left[q_{n+1} = j | q_n = i\right] \tag{7}$$

It represents the probability that the $i+1-j$ requests are responded to between two successful arriving tasks. Apparently, when $j>i+1$, probability $P_{i.j} = 0$.

The Laplace Stieltjes transformation(LST) of A arrival interval time is:

$$A^*(s) = \int_0^\infty e^{-sx} a(x)dx = \frac{\lambda}{\lambda + s} \tag{8}$$

The LST of service time B is:

$$B^*(s) = \int_0^\infty e^{-sx} b(x)dx = \frac{\lambda}{\lambda + s} \tag{9}$$

The remaining service time is the time interval from any point among service time(one arriving point of Poisson stream) to the service finishing time, and it can be denoted by B_+; the passed task service time is the service time interval from the service start time to any service time, and we denote it as $B\_$.The basic condition for the task finished before the arrival of next request is its processing time less than the request arrival interval, so its probability can be written as:

$$\begin{aligned} P_x &= \Pr ob[A > B_+] = \int_{x=0}^{\infty} P\{A > B_+ | B_+ = x\} dB_+(x) \\ &= \int_{x=0}^{\infty} \left(\int_{y=x}^{\infty} \lambda e^{-\lambda y} dy \right) dB_+(x) \\ &= \int_0^\infty e^{-\lambda x} dB_+(x) = B_+^*(\lambda) \end{aligned} \tag{10}$$

When service platforms are idle by chance, the arriving requests can be processed immediately, the probability P_y of this request processed and finished before the arriving of next request can be calculated by:

$$\begin{aligned} p_y &= \Pr ob[A > B] = \int_{x=0}^{\infty} P\{A > B | B = x\} dB(x) \\ &= \int_{x=0}^{\infty} \left(\int_{y=0}^{\infty} \lambda e^{-\lambda y} dy \right) dB(x) \\ &= \int_0^\infty e^{-\lambda x} dB(x) \\ &= B^*(\lambda) \end{aligned} \tag{11}$$

When the arrival task queue is not empty, there will be following cases. If there are some tasks completed between two successful arriving internals, service platform will take one task in that nonempty queue, and if this new task is still completed before the next arriving request, the service platform will take another new task in that nonempty queue, and the like; when the queue is empty, there are new arrival requests. So, under the circumstances that there are enough tasks to take, the finishing probability of k(>0)tasks at the service platform before the next request arrival can be calculated according to (10) and (11):

$$P_{z,k} = B_+^*(\lambda)(B^*(\lambda))^{k-1} \tag{12}$$

At steady state, let W(x) and W*(s) be CDF and LST of service waiting time W, so

$$\begin{aligned} W^*(s) = &\sum_{k=0}^{k=m-1} \left(\sum_{j=0}^{2m} \pi_j p_{j,i} \right) \\ &+ \sum_{k=m}^{k=2m} \left(\left(\sum_{j=0}^{2m} \pi_j p_{j,i} \right) (1 - s/\lambda)^{k-m} \right) \end{aligned} \tag{13}$$

Among them, $\pi = [\pi_0, \pi_1, \pi_2, \ldots]$ is steady state probability.

5 Processing time based on processor mode

The rapid development of IT technology have made the computer break away from single core era, and enter the

dual-core and quad-core even more advanced multi-core era. These multi-processor devices also cause the strategy difference of processor responding tasks. In cloud computing virtual machine technology, there are two processing strategies of task processing of the processor(Calheiros et al. [18]): Space Share and Time Share [12]. The Space Share of virtual machine is that the virtual machine processes the tasks according to tasks arriving time, and the Time Share refers to allocating all virtual machine requests to each processor; The Space Share of tasks means treating multiple processors as one processor, so one task will be allocated to multiple processors; the Time Share of tasks means letting one time slot of each processor responds to the requests from two virtual machines simultaneously. So, the Space Share of the virtual machine means, at any time, there is only one virtual machine processing task, and then the tasks will be processed one by one; and the Time Share of virtual machine means, at any time, there are multiple virtual machines responding to task requests, which leads to all the tasks responded to at the same time [13]. Therefore, there are four ways of strategy combinations. In view of dual-core and two virtual machines, four strategy combinations are just represented like Fig. 3: CPU Space Share - task Space Share (shown in Fig. 3a), CPU Space Share - task Time Share (shown in Fig. 3b), CPU Time Share - task Space Share (shown in Fig. 3c), CPU Time Share - task Time Share (shown in Fig. 3d). To explain the differences between these four processing strategies, we take one dual-core computer device as an example. We assume cloud computer service request two virtual machines and receive user requests at the same time, and the service requests of VM1 consist of t1∼t4, and the service requests of VM2 consist of t5∼t8; different processing strategies have differences in time allocation methods and tasks performing orders in processors.

Assume the virtual machine received p cloud tasks(Cloudlet), each task includes $r(j), j = 1, 2, \ldots, p$ operating instructions, main host has n processing units, each processing unit has *cap(i)*, $i = 1, 2, \ldots, n$ processing length, $cores(j)$ denotes the processor number which tasks need, and the capability of main host can be represented below.

$$cp = \frac{\sum_{i=1}^{n} cap(i)}{n} \tag{14}$$

So, at the Space Share mode of the virtual machine, the consuming time ET of virtual machine to perform task $j$is:

$$ET(j) = \frac{r(j)}{cp \times cores(j)} \tag{15}$$

The waiting time WT of task $j$is:

$$WT(j) = \sum_{k=1}^{j-1} ET(k) \tag{16}$$

So, at the Space Share mode, the finish time of task j is:

$$\begin{aligned} FT(j) &= WT(j) + ET(j) \\ &= WT(j+1) \\ &= \sum_{k=1}^{j} \frac{r(k)}{cp \times cores(k)} \end{aligned} \tag{17}$$

For co-processing the multiple tasks in the Time Share mode, the waiting time of task j is 0, and the consuming time ET of the virtual machine to perform task can also be represented by formula (16), merely the computing way of the processing capability of main host is not quite the same.

$$cp = \frac{\sum_{i=1}^{n} cap(i)}{\max\left(\sum_{j=1}^{p} cores(j), n\right)} \tag{18}$$

So, in Time Share mode, the finish time of task j is:

$$\begin{aligned} FT(j) &= ET(j) \\ &= \frac{r(j)}{cp \times cores(j)} \end{aligned} \tag{19}$$

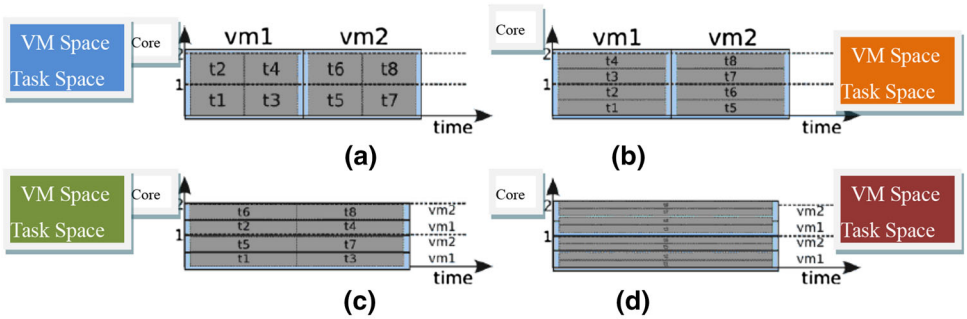


Fig. 3 Comparing two processing strategies

Through the previous comparisons, we can see that, in Space Share mode, users would have the sense of service waiting, but when the service get the cloud response, it will be finished quickly. Meanwhile, in Time Share mode, users have a quick requests response, but there will be a longer processing time of one single task. In fact, whatever the mode is, the real time which a processor used to respond to and process the access requests are equally ET; the only difference is the different consuming time due to the different allocating resources.

6 Network delay time based on network topology

Based on colossal computer networks, cloud computing has numerous network transmission and communication devices and connects with local end clients all over the world; therefore, the network topology becomes the cornerstone of cloud service system [14]. The network information transmission delay directly affects the service quality of the entire service system, but for cloud service system builders, users and administrators, the actual physical topology of network is neither the object of their concern, nor the contents that they improve network to reduce network delay with their existing ability. So in most cases, network delay time can be used as an exogenous characteristic of cloud service network only, and reduce network delay through routing algorithms in some possible circumstances [15]. Usually the following delay matrix is used to represent the delay state in the information transmission of each network main node.

$$L = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1r} \\ l_{21} & l_{22} & \cdots & l_{2r} \\ \cdots & \cdots & \cdots & \cdots \\ l_{r1} & l_{r2} & \cdots & l_{rr} \end{bmatrix} \tag{20}$$

Among them, l_{ij} represents the transmission time needed for transmitting information or data from network node i to node j, that is, after the transmitting instruction is issued at node i, the information will be transmitted to node j after delaying l_{ij} ms, as in Fig. 4. This matrix represents the data transmission delay information among r major network entities, such as Asia, Africa and Europe and other major regions of the world, and reflects the region entry port speed and export port speed, as is shown in Fig. 4.

Figure 4 shows the data delay phenomena caused by cloud service network topology; that is, the physical property of network topology brings about the delay between sending and receiving information of cloud clients, which should also be included in the service quality of cloud service provider directly (Fig. 5).

The description of the network topology usually includes all the information of every network node [16]. These nodes in the cloud environment include hosts, data center and cloud agencies, and so on. In BRITE, the node information consists of node index, row and column numbers, out-degree and in-degree, etc.; the edge information includes edge number, starting node, ending node, Euclidean distance, transmission delay and bandwidth, etc. The information included provides the direct basis for the delay calculation of each network node.

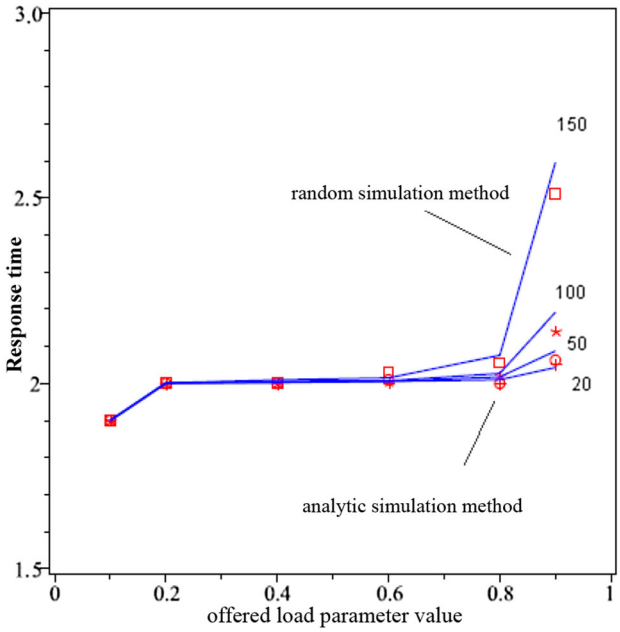


Fig. 5 contrasting diagram of simulation results

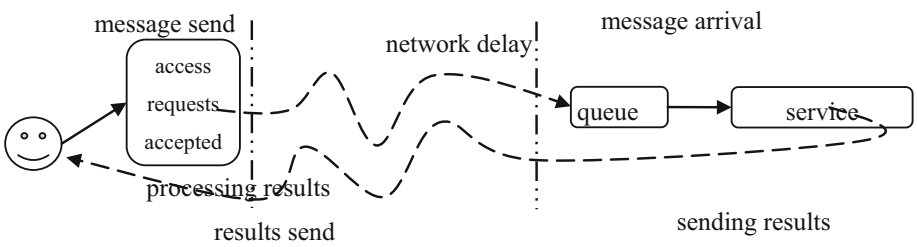


Fig. 4 diagram of network transmission delay

7 Cloud service quality assessment model based on response time

One important indicator measuring service system in classical queuing theory is the sojourn time W_s that customers stay in a system, including customer queuing waiting time W_q and service time $1/\mu$. For cloud system, due to the network topology, there is an additional indicator: network transmission delay time L; because network delay time is an exogenous variable, treated as a constant. Based on the previous discussion and formula(1), the response time of cloud service system is:

$$\begin{aligned} W &= E\left(\frac{1}{n}\sum_{i=1}^{n} W_s^i + T_i\right) = \frac{1}{n}E\left(\sum_{i=1}^{n} W_s^i\right) + 2L \\ &= \frac{1}{n}\left(E\left(\sum_{i=1}^{n} W_s^i\right) + 2\sum_{j=1, j\neq i}^{j=r} l_{i,j}\right) \\ &= \frac{1}{n}\left(E\left(\sum_{i=1}^{n} W_q^i + \mu\right) + 2\sum_{j=1, j\neq i}^{j=r} l_{i,j}\right) \\ &= \sum_{k=0}^{k=m-1}\left(\sum_{j=0}^{2m}\pi_j p_{j,i}\right) \\ &\quad + \sum_{k=m}^{k=2m}\left(\left(\sum_{j=0}^{2m}\pi_j p_{j,i}\right)(1-s/\lambda)^{k-m}\right) \\ &\quad + \frac{r}{cp \times cores} \\ &\quad + \frac{2}{n}\sum_{j=1, j\neq i}^{j=r} l_{i,j} \end{aligned} \tag{21}$$

Among them, l_{ij} represents the network delay time needed for transmitting information or data from client terminal to cloud processing terminal. It can be seen that the system response time is composed of three portions: queuing waiting time, service processing time and network delay time.

8 Conclusion

There are two ways of the verification of the numerical simulation of the service level evaluation model of this cloud service system: random simulation method and inference simulation method. The random simulation method generates the random data conforming with specific distribution feature by Monte Carlo method, to simulate the operation status of the service system, and finally describes the whole instance values of the state variables of the service system which needs inspecting by statistical variable like mean value and variance etc. Random simulation can simulate the uncertainty of the real world very well, verify the performance of the extreme points when the system distribution permits, and conduct the all-round service inspection. But because random simulation has the random feature, each simulation produces different results, and in view of statistics, they are approaching the ideal values of state variable infinitely, which leads to the value errors. Meanwhile, because random simulation gets the persuasive data when enough simulations have been performed, simulating enough times is required, which will consume a great deal of time [17].

Inference simulation performs inference simulation using the statistical features of random variable directly instead of simulating the statistical features of the random variable. That kind of simulation perform the simulation directly using derived analytic formula, getting the computing value of the state variables, instead of statistical value, which eliminates the random errors, and reduces the computing workload of the system greatly. But statistical feature could express the information of most sample points; at the same time, it will also neglect the information of some individual extreme points so that it can lead to an incomplete system inspection, and the inability to find problems that happened in the extreme circumstances.

Since cloud computing involves simulating a lot of hardware resources and user accesses, it requires a great number of computations [18]. Keeping using random simulation method, the demand for the experiment devices will be extraordinary high, and the time it consumes would be beyond imagination. Therefore, in this article, inference simulation method will be adopted to analyze cloud service system in the follow-up studies, and, for this reason, the validity of the simulation results of inference simulation method should be verified first.

Based on the analytic value of system response time given in (21), the validity of the evaluation model of the service system has been verified by the contrast of this analytic value with the random simulation results which adopts the same assumption [19]. Simulate the discrete event of cloud service using the Artifex simulation engine based on Petri network, and compare the mean values of the results of these two methods in the cases of different virtual machines. The results are shown below:

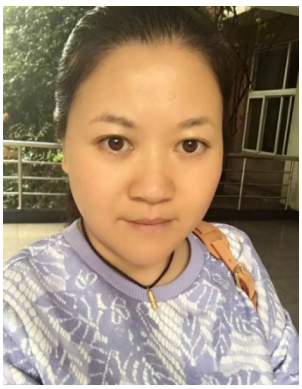
Therein,the poly-lines represent the mean values of system response time of all the access requests with the different parameter values when random simulation method has been adopted, and the marker points show the system response time at each view point of Markov chain by analytical formula simulation. The corresponding numbers on the four lines represent the number of simulation virtual machines. In that figure, it shows that the response time of the two methods is very close. Working off the random errors of random simulation method, it can be concluded that the evaluation model of cloud system raised in this article is effective, and

the simulation of cloud service system employing analytic simulation methods is feasible and effective.

Acknowledgements This work was supported by Soft Science Research Program of Zhejiang Province (No.2016C25G2080022), Philosophy and Social Sciences Key Research Base of Hangzhou City-Electronic Commerce and Network Economy Research Center of Hangzhou Normal University (No.2015JD30), and Education Department Scientific Research Project of Zhejiang Province (No.Y201432184).

References

1. Vani, B., Priya, R.C.M.: A survey on the security issues in cloud computing. Int. J. P2P Netw. Trends. Technol. **11**, 16–19 (2014)
2. Chard, K., Caton, S., Rana, O., et al.: Social clouds: a retrospective. IEEE Cloud Comput. **2**(6), 30–40 (2015)
3. Du, N.H., Huang, H.L., Li, L.F.: Can online trading survive badmouthing? An experimental investigation. Decis. Support Syst. **56**(6), 419–426 (2013)
4. Hakiri, A., Gokhale, A., Berthou, P., et al.: Software-defined networking: challenges and research opportunities for future internet. Comput. Netw. **75**, 453–471 (2014)
5. Montes, Jesús, Sánchez, Alberto, Memishi, Bunjamin, et al.: GMonE: a complete approach to cloud monitoring. Future Gener. Comput. Syst. **29**(8), 2026–2040 (2013)
6. Bastug, E., Bennis, M., Zeydan, E., et al.: Big data meets telcos: a proactive caching perspective. J. Commun. Netw. **17**, 549–557 (2015)
7. Dinh, H.T., Lee, C., Niyato, D.: A survey of mobile cloud computing: architecture, applications, and approaches. Wirel. Commun. Mob. Comput. **18**, 1587–1611 (2013)
8. Lin, C.H., Liu, D., Pang, W., et al.: Sherlock: a semi-automatic framework for quiz generation using a hybrid semantic similarity measure. Cogn. Comput. **7**(6), 667–679 (2015)
9. Li, Q., Yang, Q., He, Q., et al.: Profit-maximizing virtual machine provisioning based on workload prediction in computing cloud. KSII Trans. Internet Inf. Syst. **9**, 4850–4966 (2015)
10. Gotelli, N.J., Wener, U.: Statistical challenges in null model analysis. Oikos **121**(2), 171–180 (2012)
11. Ghosh, R., Longo, F.: Scalable analytics for IaaS cloud availability. IEEE Trans. Cloud Comput. **2**(1), 57–70 (2014)
12. Bhanu, Kaushik, Honggang, Zhang, Xinyu, Yang, et al.: Providing service assurance in mobile opportunistic networks. Comput. Netw. **74**, 114–140 (2014)
13. Thijs, Baars, Ravi, Khadka, Hristo, Stefanov, et al.: Chargeback for cloud services. Future Gener. Comput. Syst. **41**, 91–103 (2014)
14. Jararweh, Y., Jarrah, M., Kharbutli, M., et al.: CloudExp: a comprehensive cloud computing experimental framework. Simul. Model. Pract. Theory **49**, 180–192 (2014)
15. Su, Q., Chen, L.: A method for discovering clusters of e-commerce interest patterns using click-stream data. Electron. Commer. Res. Appl. **14**(1), 1–13 (2015)
16. Pichel, J.C., Rivera, F.F.: Sparse matrix-vector multiplication on the Single-Chip Cloud Computer many-core processor. J. Parallel Distrib. Comput. **73**, 1539–1550 (2013)
17. Su, Q., Huang, J.J., Zhao, X.D.: An information propagation model considering incomplete reading behavior in microblog. Phys. A **419**(2), 55–63 (2015)
18. Calheiros, R.N., Ranjan, R., Beloglazov, A., et al.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resources provisioning algorithms. Software **41**(1), 23–50 (2011)
19. Paula, P., Fazendeiro, P., Augusto, C., et al.: Ambiente Colaborativo para Avaliação de Cadeias de Abastecimento Collaborative Environment for Supply Chain Ass-essment [J]. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação **12**, 1–15 (2013)
20. Xu, Z., Mei, L., Liu, Y., Hu, C., Chen, L.: Semantic enhanced cloud environment for surveillance data management using video structural description. Computing **98**(1–2), 35–54 (2016)



ZheXi Yang was born in 1974, and graduated in Shanghai University of Finance and Economics. She has a doctor degree, and her major is Management Science and Engineering. Now she works as a teacher in China Jiliang University, and her research area is: Management Information Systems, Equilibrium Optimization & Standardization.

Wei Liu was born in 1961, and graduated in National Defense University. He has a doctor degree, and his major is army command and control. Now he works as a teacher in China Jiliang University, and his research area is: System Engineering.

Duo Xu was born in 1996, and she is studying in China Jiliang University. She has a bachelor degree, and her major is Standardization. Now she is a student in China Jiliang University, and her research area is: Standardization.