

A Complex Network Structure Evaluation Model Based on Cloud Computing

Xiaohui Ji and Jie Xie

*School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang 471023, China
78821389@qq.com*

Abstract

Cloud systems are difficult to assess due to the relevance of modeling and its large-scale, complex structure, highlighting the dynamic and powerful layer. To solve this problem, the cloud system evaluation model is proposed based on queuing Petri network. It can efficiently model and simulate complex cloud system due to its powerful combination of quantitative evaluation and behavior description of the theoretical characteristics of queuing theory and Petri nets. On the basis of the model, we further increase the quantitative analysis and evaluation system cloud systems, and simulation of cloud systems, massively parallel dynamic service environments. Experimental results show that, queuing Petri network can effectively reflect the architectural features of various cloud computing system performance perspective, services.

Keywords: *Cloud Computing, Queuing theory, Evaluation model, Petri network*

1. Introduction

Cloud computing has rapidly developed in the fields such as government, economy and scientific research because of its advantages involving low cost, high expansibility, simplicity and so forth. Various cloud systems with different architectures, technologies and goals have appeared and show fast popularization and application [1].

It is one of the key factors for successfully implementing cloud computing to find out the ways for efficiently and accurately evaluating cloud architectures considering diverse cloud systems with miscellaneous technologies. Essentially, cloud computing is a kind of new service mode, in which, the service types and characteristics required by different service fields may significantly differ from each other. Therefore, it is difficult to obtain a kind of architecture or scheme of cloud computing which can provide optimal service and adaptability for all fields. Thus, it is more likely to result in problems such as high cost of cloud computing, low quality of services and waste of resources to deploy and construct cloud computing blindly in the absence of evaluation and prediction. Consequently, it fails to effectively satisfy practical service demands and performance requirements.

2. Related Work

In order to evaluate cloud computing system, traditional evaluating techniques such as simulation systems (CloudSim, Emusim, koala, icancloud, etc.), prototype system and static mathematical model have been employed in many investigations. However, the characteristics such as huge public scale, complex structures, outstanding dynamics, and strong associations among layers may induce defects including poor capability of index quantification, high evaluation cost, low simulation degree, limited scale, etc. in practical evaluation process. Generally, the frequent lack of adequate theoretical support of simulating evaluation system makes it difficult to effectively simulate the new characteristics and details of features. Especially, the characteristics of cloud computing

including large amount of users, complicated behaviors and dispersed distribution bring more difficulties to simulate cloud computing environments. Moreover, prototype system calls for high evaluation cost, and is difficult to be changed once achieved; therefore, it impedes the comparative evaluation on multiple architectures and platforms. As the scale and structure of cloud system are increasingly complex, the formalized model of the cloud system is increasingly huge and the complexity increases exponentially. Traditional methods based on mathematical derivation and theoretical explanation also fail to solve the problem of the explosive growths of data quantity, data amount and calculation amount existing in the analytical process of the formalized model of cloud systems. Stochastic queuing Petri net [2], as a kind of powerful graphics-oriented modeling language, not only can describe complex architectures of cloud computing, but also is able to dynamically analyze all kinds of behaviors of cloud computing system strongly. It inherits and develops the advantages of queuing network model and stochastic Petri net. Therefore, in addition to quantitatively modeling the operation performance of sources, it also can depict the dependency relationship among multilayered systems considering the characteristics of cloud system involving diverse application service, complex service behavior and superimposed hierarchical structures [3-4]. Therefore, Petri net is the first choice for system modeling because it is applicable for the evaluation and prediction of cloud computing both in terms of theoretical support and quantitative analysis.

According to the aforementioned problems, an evaluation model of cloud system on the basis of Petri net was proposed. Based on the system features of “Chinese cloud” of the *State Key Laboratory of High-end Server & Storage Technology*, a detailed and complete simulation modeling was performed on typical cloud architecture using this model by combining the characteristics of queuing Petri net theory. On the basis of this, the dynamic service performance provided by cloud computing was quantitatively analyzed. Experimental results revealed that the evaluation model was capable of reflecting the service characteristics and performance features of various kinds of cloud architectures. Meanwhile, the results of quantitative analysis were valuable references for promoting further development and application of cloud computing.

3. System Architectures of Typical Cloud

Figure 1 illustrates the basic working process, during which, cloud platform deals with user's tasks and works in accordance with the characteristics of the “Chinese cloud” project in terms of basic architectures. Dispatchers correspondingly distribute the works and tasks which reach the underlying resources. The obtainment of works and tasks by a computational node calls for a corresponding treatment, during which, the communication among computer nodes through the network is needed so as to interact computing information. Meanwhile, the computing nodes are also asked to communicate with storage nodes in order to fulfill the I/O operation of data, the communication of which can be accomplished through network or directly through SCSI, *etc.* The data of final results are returned to users via the network. Additionally, in some cases, the transfer operation of data among storage devices is conducted through the network as well.

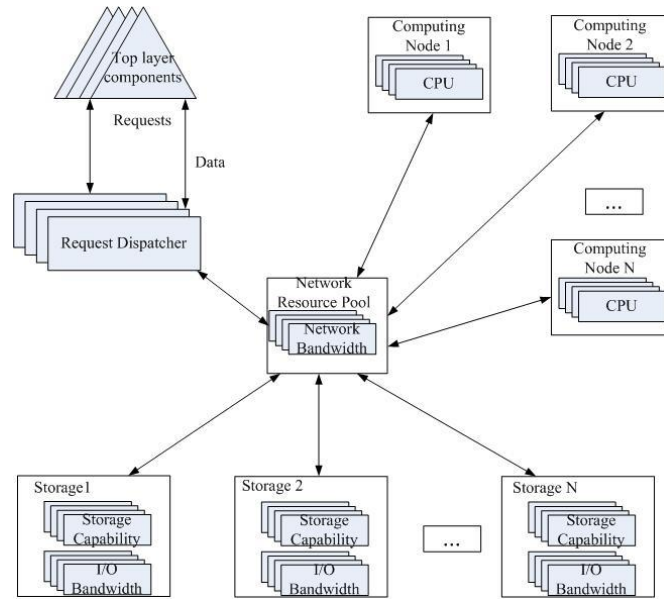


Figure 1. Task Processing on Cloud Platform

Based on the current cloud systems in the world, the Chinese cloud system proposed three representative and universal architectures of cloud system as the alternative scheme of the system architecture of Chinese cloud by combining the characteristics of the project. They are distributed, centralized and hybrid architectures of cloud system, as shown in Figures 2, 3 and 4, respectively, and described in details as follows.

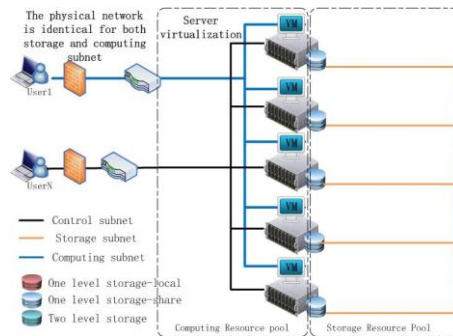


Figure 2. Distributed Architecture of Cloud System

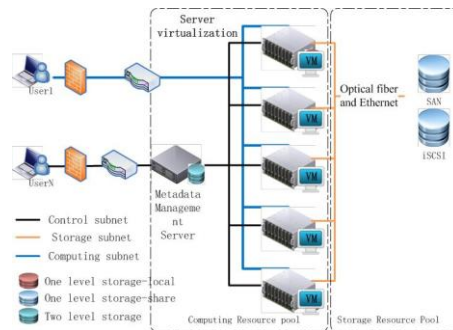


Figure 3. Centralized Architecture of Cloud System

Figure 2 shows the distributed system architecture, in which, virtual machines are placed on many computing nodes. Therefore, a computing resource pool is constituted, and represented and described using a computing subnet in the layered queuing networks. These peer-to-peer independent computing nodes are connected via the computing subnet. Meanwhile, all computing nodes possess their own incidental storage devices which are organized in a sharing form. The data are shared through the storage subnet which is used for the indication and description in the layered queuing Petri net. It is noteworthy that the storage and computing subnets are identical physically. Consequently, the competition of systems on network resources is induced, and expressed and illustrated using the network subnet in the layered queuing Petri net. Network resources may be the bottlenecks controlling the evaluation index such as performance and expansibility of distributed cloud system under high loads. Therefore, the competitive characteristics on network resources need to be reflected in subsequently layered queuing Petri net.

Figure 3 illustrates the centralized system architecture. As shown in the figure, virtual machines are laid on multiple computing nodes and therefore a computing resource pool is formed, which is denoted and expressed using a computing subnet in the layered queuing networks. There are one or several server nodes of metadata mainly providing metadata service on computing, storage, *etc.* in a centralized form. Metadata servers and computing nodes are connected through computing subnet. Therefore, the cloud system is required to obtain metadata information on computing and storage of virtual machine through metadata service in advance of providing various services of computing and data operation subsequently. In addition, the centralized architecture of cloud system also provides centralized storage service. By using optical fiber or ordinary Ethernet, the computing subnet is connected on storage pools such as SAN and iSCSI. The interconnection of storage devices and interaction of data are fulfilled using the storage subnet inside the storage pools, for example, SAN, NAS, iSCSI and so on. Therefore, the storage and computing subnets are independent in the centralized system architecture physically, and this favors the concurrent processing of storage and computing information. They are expressed and depicted utilizing the storage and computing subnets in the layered queuing networks. However, the metadata server, as the center of computing and storage, may be the performance bottleneck under high loads. Thus, it is required to reflect the independence of the subnets and the potential bottleneck problems of metadata server involving performance and expansion in subsequently layered queuing Petri net.

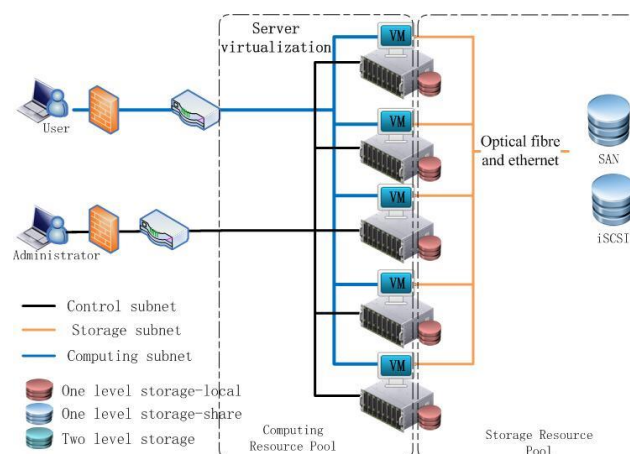


Figure 4. Hybrid Architecture of Cloud System

3. Evaluation Model of Cloud Architecture Queuing Petri Net

Based on the mentioned analysis on the characteristics of the alternative architecture scheme of China Cloud, the queuing Petri net models of distributed centralized and hybrid architectures of cloud system are constructed. All architecture models of cloud system contain corresponding subnets of storage, network and computing and thus constitute the layered queuing Petri net model of China Cloud system.

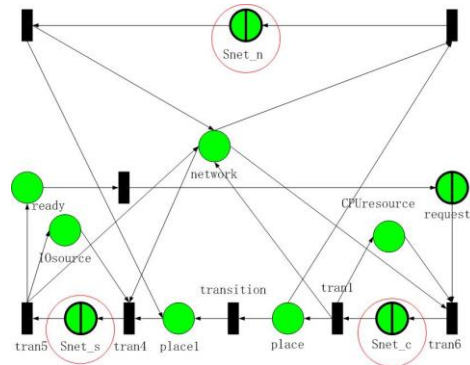


Figure 5. Distributed Architecture of Cloud System

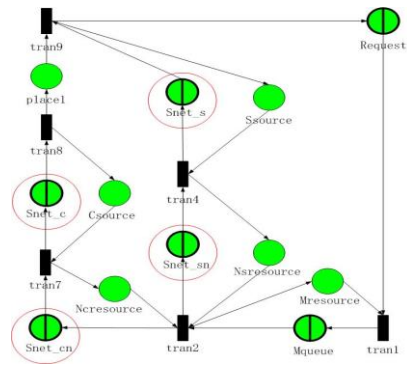


Figure 6. Centralized Architecture of Cloud System

The Petri net model of distributed cloud system is displayed in Figure 5. Confined to the length of the paper, merely most important places and transitions are listed.

Request is a queue of the queuing and describes the work and task requests sent by users and upper components, in which, queuing time refers to the think time between works and tasks. The think time can be distributed exponentially according to practical system. *CPUresource* is a common position and mainly expresses the amount of computing resources. The resources quantity varies with the changes of the hardware devices, in which, Token quantity refers to the quantity of computing resources. *Snet_c* is a queuing subnet which depicts the tasks of cloud system queuing in computing subnet and performs relevant calculation of the tasks. *Snet_s* is a queuing subnet, describes the queuing of tasks in storage subnet and processes the tasks. *IOsource* is a common position which represents I/O source with the most of bandwidth resources. *Network* is a common position and depicts the network bandwidth resources in the current architecture of cloud system, in which, Token quantity represents the quantity of network resources. *Snet_n* is a queue of the queuing and refers to the transmission in a queue of the tasks in network subnet, in which, the network sources are competed by computing and storage subnets. *IOQ (remote)* refers to a queue of the queuing and indicates that the data request of the tasks reach remote storage devices, through which, the data are operated in a queue. The queues present that the tasks operate data in a queue on the remote storage devices, in

which, corresponding parameters refer to the time and distribution of I/O processing. Concrete collocations are set based on simulations.

The main transitions in the model are expressed as follows. Tran 6 indicates that the tasks occur, enter into the virtual machine and conduct calculation. The trigger of this transition calls for corresponding support of computing resources, one of which is occupied by the tasks. Tran 3 illustrates that task data are not stored in local storage devices, triggers the transmission of the network data in next stage and obtains data from remote storage devices. This transition can be caused by corresponding network resources, one of which is occupied by the tasks. Meanwhile, this transition possesses weight of trigger. Tran 4 explains that task data are stored in the local storage devices and sources the operation of local data in next stage. The occurrence of this transition requires corresponding support of I/O resources, one of which is occupied by the tasks as well. This transition also shows the weight of trigger. Tran 10 demonstrates that the tasks reach remote storage devices and results in the operation of remote data in next stage. This transition can be induced by corresponding I/O resources, one of which is occupied by the tasks.

The queuing Petri net model of distributed architecture of cloud system in Figure 5 shows that the network resource in the place (network) has been the key resources of the whole architecture of cloud system. The network resource is required either by the transmission of computing information or the operation of non-local data during the calculation process through the virtual machine by one task.

The Petri net model of centralized architecture of cloud system is plotted in Figure 6, in which, merely most important places and transitions which are different from those in Figure 5 are listed.

Mresource is a common position and refers to the metadata server resource of the centralized cloud system. The resource quantity determines the performance of the whole system to a great extent owing to the centralized resources. *Mqueue* is a queue of the queuing and describes the queue processing of the tasks on metadata server. It is required to firstly obtain corresponding virtual machine and data information through metadata server in the case of the arrival of new tasks. *Snet_sn* is a subnet of the queuing and indicates that the task transmits relevant data in storage subnet in a network queuing. *Msresource* is a common position and refers to the network resources of the storage subnet in the centralized cloud system. The application of a centralized storage results in the mutual independence of storage and computing subnets. *Snet_s* is a subnet of the queuing, describes the queue of data processing of tasks in storage subsystem and calls for the support of relevant storage resources. *Snet_cn* is also a subnet of the queuing and describes the computing operation of relevant tasks in network queuing in a computing subnet. *Snet_c* is a subnet of the queuing, depicts the task computing queuing of tasks in computing subsystem and requires the support of relevant computing resources. *Csource* is a common position, which illustrates the computing resources in current computing subsystem. The resource quantity is dynamically adjusted during simulation process. The transitions are not given in detail because they are similar to those corresponding to Figure 5.

The Petri net model of centralized architecture of cloud system in Figure 6 indicates that the metadata server resource in the queues of queuing (Mqueue) is the key resource in the whole architecture of cloud system. One task is required to obtain the information of computing and storage through metadata server in advance. In addition, the proportions of computing or storage resources needed by the same task are different in various types of applications. This is achieved by adjusting the Token quantity of Csource and Ssource in subsequent simulation experiments.

The Petri net model of hybrid architecture of cloud system is shown in Figure 7. The main places and transitions are similar to those in Figures 5 and 6 and are not described in detail as space is limited.

The Petri net model of hybrid architecture of cloud system in Figure 7 presents that the hit rate of local replica of computing nodes is essential for improving the performance of the whole system. High hit rate of local replica can save the consumption of the network resource in resource pools and thus improves the performance of data operation.

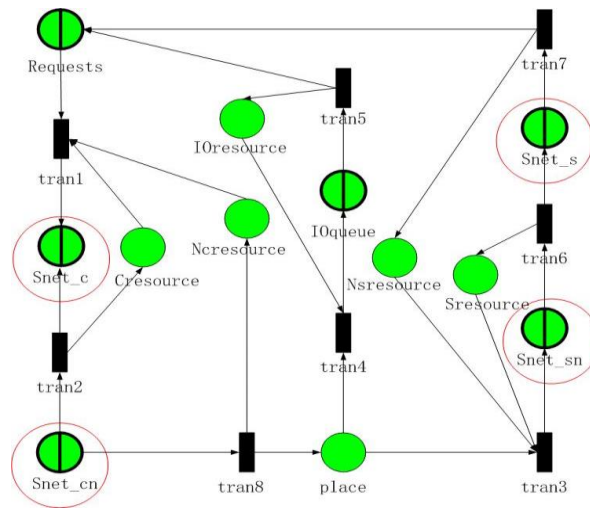


Figure 7. Queuing Petri Net Model of the Hybrid Architecture of Cloud System

4. Analysis of Simulation Results

Currently, there are many kinds of software and tools used for the automatic calculation of mathematical models at home [5-8] and abroad [9-11]. According to the aforementioned characteristics of queuing Petri net, the QPME tool [12, 13] was adopted to simulate the layered queuing Petri net model of cloud system in this investigation. In this way, the concrete manifestations of evaluation indexes such as performance and expansibility under different system architectures and allocations were explored. Therefore, it offered helpful reference and advice for deploying and constructing cloud systems.

In order to evaluate the service characteristics and performances of China Cloud under different scales and loads, the load was further divided into 3 types by the model. $X\% : Y\%$ refers to that the data of $X\%$ can be acquired through local storage system rather than the network in distributed or hybrid system architecture, while the data of $Y\%$ need to be read through the network. All data in centralized system architecture are obtained through the remote call of data network. Therefore, they are compute-intensive (Computing), data-intensive (Data), and calculation data balanced-type (Balance) architectures according to loads. The performances of the 3 types of cloud system architectures under different loads are displayed in Figures 8, 9 and 10, respectively.

Figure 8 presents the changes of the task throughput in the distributed cloud system under different loads. Simulation results showed that the throughput of the whole system increased with the increase of loads. Simulated data also presented that as the range of the working set continuously increased, the data directly obtained through local storage device decreased because more data needed to be acquired from remote storage device via network transmission. All these increased the overall postpone of the system and thus reduced the throughput. However, the network was the performance bottleneck of the whole cloud architecture and thus the throughput decreased rapidly as the data passing through the network constantly increased to approximate or exceed network process

ability. As can be seen, the throughput of the state of 80% : 20% decreased by 36.9% compared with that of the state of 20%:80%.

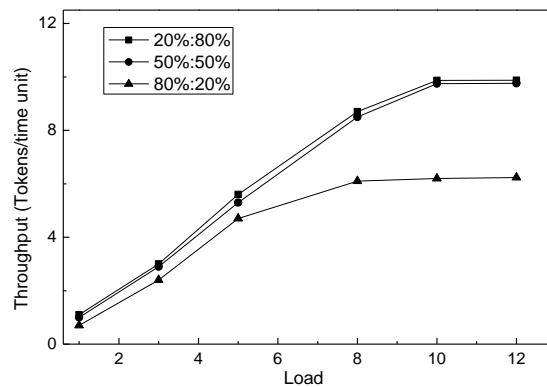


Figure 8. Cloud System under Different Loads

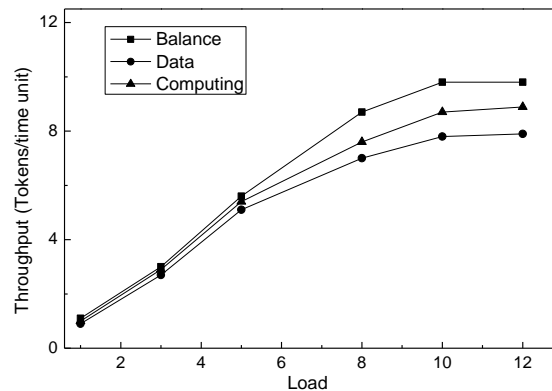


Figure 9. Centralized Architecture of Cloud System under Different Loads

Unlike the distributed architecture of cloud system, the simulation data in Figure 9 indicated that the variations of application types resulted in unbalanced resources of the cloud system. The subsystem with inadequate resources was likely to be the performance bottleneck of the whole system, and thus reduced the throughput of the system. As shown, the throughput of the system in balanced type was 9.803. The final throughput was merely 7.894 in data-intensive application owing to the great control of storage performance showing a decrease of 19.5% compared with the ideal balanced state. However, the cloud system presented an overall throughput of 8.88 which was between the former two ones under the compute-intensive application state. It was because that under this model, although the storage system was underutilized, the storage resources exerted more significant influence on the performance of the whole system than computing resources.

Figure 10 demonstrates the performance of hybrid architecture of cloud system. Simulation data showed that as the hit rate of local replica data decreased, the overall throughput of the system decreased gradually. However, the throughput was basically stable in case of the hit rate exceeding 8, during which, all data could not be acquired from local replica (0% : 100%). Finally, in case of the hit rate of the replica data exceeding 20%, its increase insignificantly affected the increase of the overall throughput of the system.

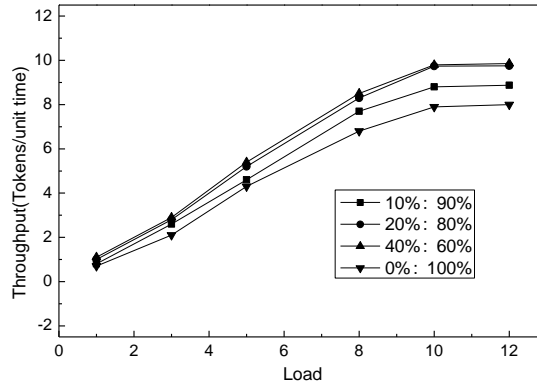


Figure 10. Throughput of the Hybrid Architecture of Cloud System under Different Loads

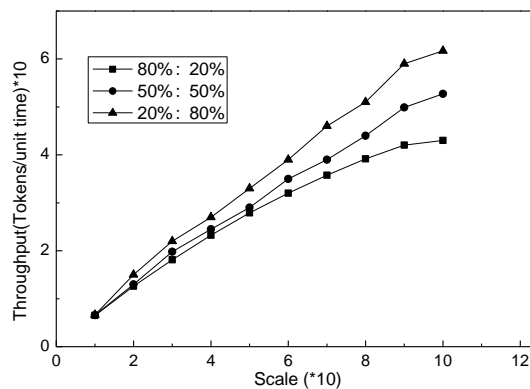


Figure 11. Throughput of the Distributed Architecture of Cloud System under Different Scales

The experimental results in Figure 11 showed that the expansibility of the distributed cloud system was controlled by network resources. With the increase of the system scale, the entire performance of the cloud system continuously increased. As the scale expanded from 10 to 40, the throughput of the system increased from 6.51 to 23.24 showing a 3.56 times of increase and a basically linear expansion in the case of adequate network resources. However, as the scale enlarged, the network resources gradually became scarce, especially, when the resources quantity needed by the system approximated or reached the amount provided by the network resources, the cloud system presented a sharply declined expansibility. Furthermore, the simulation data also indicated that the smaller the proportion of obtaining remote data via network, the better the expansibility of the system and the more insignificant control of network resources in the distributed architecture of cloud system. It could be observed that the distributed architecture of cloud system could more favorably adapt to compute-type application or the applications with intensively distributed working set of the data instead of those applications with large data and large range of working sets.

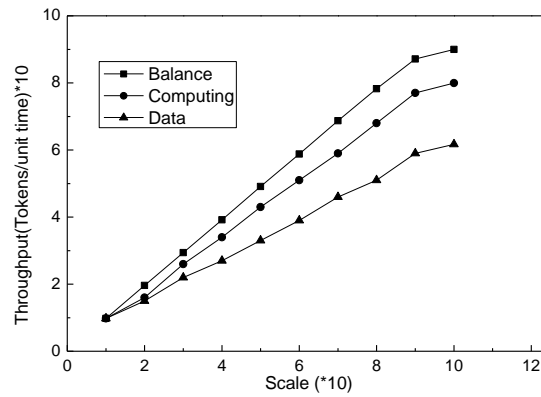


Figure 12. Throughput of the Centralized Architecture of Cloud System under Different Scales

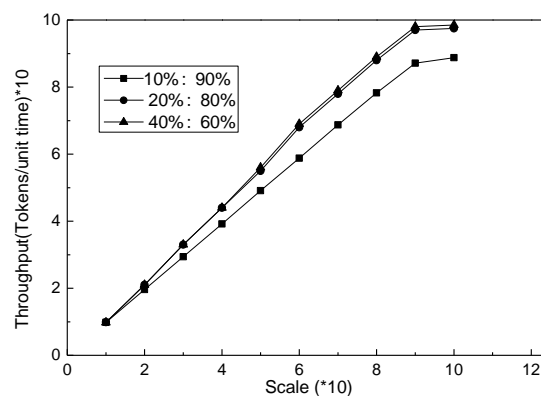


Figure.13 Throughput of the Hybrid Architecture of Cloud System under Different Scales

The results in Figure 12 presented that with the enlargement of the scale, the metadata server showed a scarce resource. The expansibility of the system decreased sharply in the case of the resource quantity required by the system approximating or reaching the ultimate service of the metadata server. The simulation data indicated that the main performance indexes such as throughput merely increased by 1.03 times as the scale of the system expanded from 90 to 100 with a 1.1 times of enlargement. However, as the scale increased by 2 times enlarging from 100 to 200, the throughput only increased by 1.002 times owing to the control of metadata server. The balanced application type usually brought more preferable throughput and expansibility to the architecture in the centralized architecture of cloud system.

The experimental results in Figure 13 presented better expansibility of the system in the hybrid cloud architecture. Compared with the former two architectures, the hybrid cloud architecture possesses more favorable throughput and expansibility especially in the case of high hit rate of local replica data.

5. Conclusions

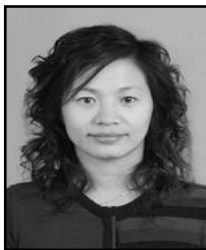
This study proposed an evaluation model of cloud systems based on the queuing Petri net and conducted layered modeling and evaluation on China Cloud system. In addition to quantitatively modeling the operation performance of resources, this model could effectively depict the architecture characteristics of complicated and layered cloud system because of its inheritance and development of the advantages of queuing net model and stochastic Petri net. Research results indicated that this model was capable of accurately describing and analyzing the relationship among various layers and component models in

cloud system owing to its favorable adaptability to the diversity and complexity of the cloud system. It effectively evaluated the performance of current typical cloud architecture by making full use of the advantages of queuing Petri net in terms of the capability of quantitative evaluation and behavioral description. Therefore, it offered the support of quantitative analysis and laid theoretical basis for deployment and on-line performance management of complex cloud systems.

References

- [1] M. A. Marsan, G. Conte and G. Balbo, A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems, *ACM Transactions on Computer Systems*, 2,93 (1984)
- [2] S. Kounev, R. Non and J. Torres, Autonomic QoS-aware resource management in grid computing using online performance models, *Proc. of 2nd International Conference Performance Evaluation Methodologies and Tools (VALUETOOLS-2007)*, October 12-15,Nantes, France, pp. 1-10.
- [3] S. M. Han, M. M. Hassan, C. W. Yoon and E. N. Huh, Efficient service recommendation system for cloud computing market, *Proc. of 2nd International Conference on Interaction Sciences Information Technology, Culture and Human-ICIS 09*, August 23-26, Seoul, Korea pp. 839-845
- [4] L. Wang, R. Ranjan and J. Chen, *Cloud Computing: Methodology, System and Applications*, CRC Press,25,110 (2011)
- [5] S. Wu and X. Bai, Interview on Petri net analysis tool on Internet, *Computer Science*, 23, pp: 27 (2014)
- [6] C. Lindeman, DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri Nets, *Performance Evaluation*, 22, 3 (1995)
- [7] A. Henryk, <http://www.winpesim.de/index.html> (2014-01-10)
- [8] E. M. Thurner. TOMSPN: A tool for modeling with SPN, *Proc. on computer System and Software Engineering*, 27, 618 (2013)
- [9] X. Zeng, H. Xiang and F. Huang. The design and implement of Petri net visibility tool, *Journal of Huazhong University of Science and Technology (nature)*, 30, 43 (2012)
- [10] J. Zhang, Z. Liu and H. Li, The study and development for Petri net graph modeling simulation system, *Journal of Beijing University of Chemical Technology (Natural Science Edition)*, 31,100 (2004)
- [11] X. Pan, T. Li and Q. Liu, A design and implement of Petri net hierarchy model modeling tool, *Computer Application and Software*, 25, 33 (2013)
- [12] S. Kounev and A. Buchmann, SimQPN - A tool and methodology for analyzing queuing Petri net models by means of simulation, *Performance Evaluation*, 63, 364 (2013)
- [13] S. Spinner, S. Kounev and P. Meier, Stochastic modeling and analysis using QPME: Queuing Petri net modeling environment v2.0, *Proc. of 33rd International Conference on Application and Theory of Petri Nets and Concurrency*, 73, 388 (2012)

Authors



Ji Xiaohui (1978-), born in Luoyang City, Henan Province , a lecturer of Luoyang Institute of Science and Technology Computer and Information Engineering Department. Her main research interests include artificial intelligence and parallel.



Xie Jie (1978-), born in Luoyang City, Henan Province , a lecturer of Luoyang Institute of educational administration department. His main research interests include semantic Web, Web service, and data mining.

