International Journal of Hybrid Information Technology
Vol.8, No.11 (2015), pp.189-198
http://dx.doi.org/10.14257/ijhit.2015.8.11.16

Research of Automatic Configuration Technology for Virtual Machines based on Cloud Computing

Xue Tao and Liu Long

College of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China
xt73@163.com

Abstract

Cloud resource managers face many problems, such as the dynamic changes of incoming load and demand elasticity of resources. From the aspect of elastic configuration management technology of virtual resources, this paper focus on how to provide quick and reliable cloud resources for users. Virtual machine resources automatic configuration management technology is proposed in this paper, the reverse reinforcement learning technique is introduced into cloud virtual resource management, configuration management process of the virtual machine is modeled as a Markov decision model. According to the running state of the application system and the dynamic changes of the input load, this technology can make an automatic decision to add or remove a number of virtual machines. Experimental results show that this technology can complete the tasks of automating configuration of virtual resource management according to the changing load, respond to end user's in a timely manner, and ensure the SLA requirements of cloud users.

Keywords: *cloud computing, automatic configuration, reverse reinforcement learning, Markov decision model*

1. Introduction

As a new mode of IT published, Cloud computing produced a lot of new type for internet service. According to different types of services, cloud computing is usually divided into IaaS (infrastructure as a service), PaaS (platform as a service) and SaaS (software as a service). Through virtualization technology, IaaS made the virtual machines share a physical resources pool and cloud computing resources providers can offer virtual machine resources to cloud computing application provider, who can deploy their applications on the virtual resource. So scheduling management technology for virtual machines is important, which will directly affect the service cloud resource utilization ability and SLA. In addition, with the scale expansion of cloud computing resources, manual management of huge numbers of resources has become very unrealistic. Now, more in need of an automatic resource management technology, which can automatically respond to load changes, and reduce the burden of management [1-2].

At present, there existing literature [3-4] study on the reinforcement learning and dynamic programming model is applied to virtual resources management in the cloud. By viewing the virtual resource allocation problems as a study object, the research literatures are from the perspective of control programming. Literature [4] focuses on the resources configuration of each virtual machine, and uses them as a study object, eventually builds a distributed configuration system for virtual machine resources, which has received good effects.

Based on the above researches, this article mainly solved the following problems: at present, the cloud resource management is facing many challenges, for example, meeting the user's SLA, allocation of resources in time with the changing load, *etc*. Based on these

ISSN: 1738-9968 IJHIT

problems, this article focused on the research in scalable extension technology and configuration management technology for virtual resource.

In this paper, we propose a new technology of automatic provision for virtual machine resource dynamically. This technology can make a dynamic decision to add or remove virtual machine nodes through a consideration of the current load and the performance index of virtual machine, such as CPU, memory utilization [5-7]. It makes the system achieve the optimal performance, so as to accommodate load, meet the user' demand, reduce services costs and improve the utilization rate of resources. In addition the automated and intelligent provision of virtual sources can effectively reduce additional costs such as the time delay result from human decision-making and manual configuration.

In order to solve the technical problem of adaptive provision decisions, we introduce the reverse reinforcement learning method which take expert strategy as the optimal study object because of that it's not easy to determine the reward function in the MDP (Markov Decision Process) for traditional reinforcement learning. Through expert demonstration, the optimal reward function is generated. Finally using apprentice learning to generate the optimal provision strategy and the corresponding action in the cluster provision strategy is the scheme of increasing and decreasing VMs dynamically [8-11].

According to the running state of the virtual cluster and the performance parameters of all VM nodes, in combination with the analysis of the adjustment behavior for cluster, we build a state transfer model based on MDP. Then the virtual cluster simulates the provision behavior demonstrated by the professional managers by using inverse reinforcement learning method. That is, from expert demonstration the cluster gains the optimal reward function and generates provision strategy. Finally, the simulation experiments verify the validity of the method [12].

2. Automatic Configuration Decisions Model for Virtual Machines in the Cloud

2.1. Analysis of Automatic Configuration Problem

Because the cloud computing application providers can dynamically rent virtual machine resources offered by cloud computing providers, therefore the purpose of this article is to provide efficient configuration technology of virtual machine resources for cloud application providers. The application scenario of this technology is to establishing an intermediate layer between cloud application providers and cloud resources provider. The scenario of configuration process for virtual machines in the cloud is as shown in Figure 1:

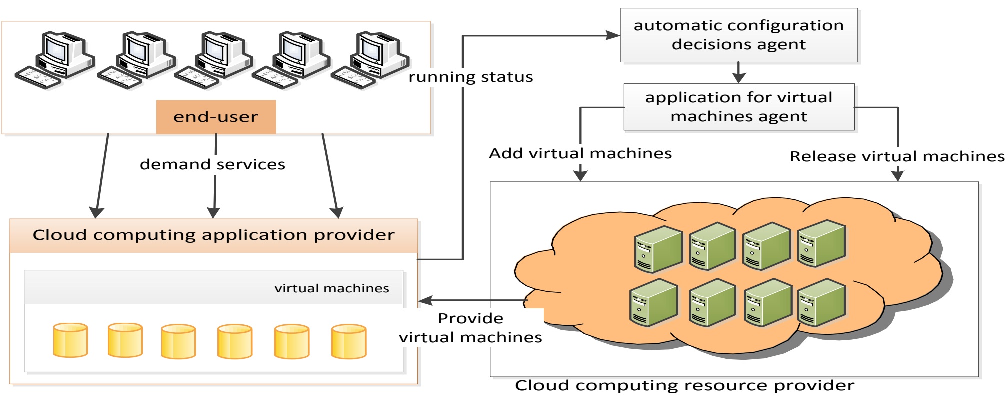


Figure 1. Scenario of Configuration Process for Virtual Machines in the Cloud

The Figure 1 shows that the main application scenario of this technology for the cloud resource providers and cloud application providers in which it provides an intermediate layer. This middle tier mainly includes two modules: automatic configuration decisions agent module and the application for virtual machines agent module. Among them, the configuration decisions agent module firstly receives application system status and the operation status of the virtual machine resources applied by cloud application providers, secondly it output configuration command to the application for virtual machines agent module based on configuration decision algorithm. The application agent module then adds and release virtual machine resources according to the received command.

By above knowable, decisions agent module is the core of this system, so the automatic configuration process for virtual machines is main the problem for this paper to solve. First of all, we regarded the automatic configuration decision process as a reverse reinforcement learning problems, which demand that the automatic configuration process is modeled as a Markov decision model [13].

2.2. Modeling based on MDP

Markov decision model (MDP) is defined by a tuple whose parameter is described below [14]:

Table 1. MDP Parameters Description

Parameter	Description
S	State space observed by agent from environment
A	Action space taken by agent
$P_{(s,a)}$	The probability in the current state S to next state s' by taking action a
R_s	Immediate reward in the current state S by taking action a
γ	Discount factor

The fundamental problem of MDP is solving the optimal strategy π which is a map of state to the action. Function is shown by the following [15]:

$$\alpha = \pi(s) \tag{1}$$

The process of establishing the automatic configuration model for virtual machines is using the markov decision model to express the process of dynamic configuration for virtual machine resources with load changing, which is as follows.

2.2.1. State Space Representation: State S is the description of the environment at some point in time, which must include all information that is grasped and used by the system and usually have an effect on the system's decision. It's usually in a more natural factor said.

State variables in this paper mainly include the numbers of running virtual machines applied by the cloud application provider. The state space of automatic configuration model is expressed as follows:

$$S = \{s_1, s_2, \ldots, s_i\} \tag{2}$$

2.2.2. Action Space Representation: Taking action in the current states, the state of the system will change correspondingly.

In this paper, when VMs cluster is under the heavy load, agent can make a decision and output commands that decreasing nodes, on the contrary, it can add nodes. Besides, we set the cluster can output action command that is adding or deleting zero nodes. We call it no-action, so the action space is defined below:

$$A = \{add^{(k)}, remove^{(k)}, no - action\} \tag{3}$$

In the last-written denominator, k denotes the numbers of adding or deleting nodes one time.

2.2.3. State Transition Function: State transfer function describes the dynamic characteristic of the system. The action of adding or removing nodes will lead to a corresponding change of physical resource and the performance of the VMs cluster. This article uses monitoring system to gain comprehensive status indicators of cluster and virtual nodes, and input these performance parameters to the agent a certain time interval. So the agent can get subsequent state of the current state at any time.

2.2.4. Reward Function Presentation: We use linear approximation of characteristic properties to describe the collection of reward function [16].

$$R_{(s)} = \varphi(s) * \omega^T, (\|\omega\|_1 \le 1) \tag{4}$$

Among them, $\varphi(s)$ as the feature attributes vector, ω as the weights matrix of feature attributes, it can not only be set by the human, but also can be computed according to the reverse reinforcement method from expert demonstration. ω^T as the transpose of a matrix.

Because optimal strategy comes in the maximum of accumulative total discount rewards, so the definition of reward function *R* directly impact on the merit rating of the selected strategy. Based on this, we must take different feature attributes into consideration and the attributes describe the limited actual reward value. For instance, when the cluster add VMs if Load increases, the cluster obtain high throughput and low delay for part of client, thus it ensures the SLA for end-users and the cloud application provider. We can give it a positive reward value; and for the server side, increasing in the number of nodes leads to greater cost consumption, we can give a negative reward value. Also it’s required to consider the CPU, memory utilization, Network load and other properties.

3. The Automatic Configuration Decisions Process for Virtual Machines in the Cloud

3.1. The Procedure Description of Automatic Configuration Decisions Process for Virtual Machines in the Cloud

We can use long-term expected rewards to evaluate the advantages and disadvantages of any strategy in MDP. By formula (1), we can get a mapping from action to strategy. The value function $V^{\pi}(s)$ is defined as the evaluation function, meaning the expectations of accumulative total discount rewards in the current state performing an action *a* [17].

$$V^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s, \alpha_0 = \alpha] \tag{5}$$

In practical solution, The intermediate variable action-value function $Q^{\pi}(s,\alpha)$ is commonly used, whose recursive form as below[18]:

$$Q^{\pi}(s,\alpha) = R(s) + \gamma \sum_{s' \in S} P^{\alpha}_{(s,s')} V^{\pi}(s') \tag{6}$$

There exists an optimal strategy $\pi^*(s)$ that can maximize the cumulative rewards In an MDP, *i.e.*, maximum $Q^{\pi^*}(s,\alpha)$. When the state - action value is known, we can get the optimal strategy π^*.

$$\pi^*(s) = \arg\max_{\alpha} Q^{\pi^*}(s, \alpha) \tag{7}$$

By the description method of linear approximation for reward function in formula (2), combined with the definition of value function in formula (5), strategyπ can be formulated below.

$$V_{\omega}(\pi) = \omega^T * E[\sum_{t=0}^{\infty} \gamma^t \varphi(s_t)|\pi] \tag{8}$$

Feature expectation $\mu(\pi)$ is defined as a standard to the similar degree between two strategies.

$$\mu(\pi) = E[\sum_{t=0}^{\infty} \gamma^t \varphi(s_t)|\pi] \tag{9}$$

We plug equation (9) into t (8), so the value function represented by features expectation is available.

$$V_{\omega}(\pi) = \omega^t * \mu(\pi) \tag{10}$$

In space of value function, we want to find value function $V_{\omega}(\pi)$ under strategy π , making it as close as possible to the value function $V_{\omega}(\pi_E)$ of expert strategy, it is expressed as below:

$$\left|V_{\omega}(\pi) - V_{\omega}(\pi_E)\right| \le \varepsilon \tag{11}$$

plugging equation (10) into (11), combined with Cauchy-Schwarz inequality: $\left|x^T y\right| \le \|x\|_2 \|y\|_2$,we can get the following,

$$\left|V_{\omega}(\pi) - V_{\omega}(\pi_E)\right| = \left|\omega^T \mu(\pi) - \omega^T \mu(\pi_E)\right| \le \|\omega\|_2 \|\mu(\pi) - \mu(\pi_E)\|_2 \tag{12}$$

By norm inequality: $\|\omega\|_2 \le \|\omega\|_1 \le 1$, formula (12) can be transformed ,

$$\left|V_{\omega}(\pi) - V_{\omega}(\pi_E)\right| \le \|\mu(\pi) - \mu(\pi_E)\|_2 \le \varepsilon \tag{13}$$

After founding the approximate expert strategy, we use the method of maximizing marginal to solve ill-posed problems of inverse reinforcement learning. The optimization problem is concluded as follows:

$$\max_{t,\omega} \tau \tag{14}$$

$$\textbf{s.t.} \quad \omega^T[\mu(\pi_E) - \mu(\pi)] \ge \tau \quad i = 1,2,\ldots,\tau - 1 \tag{15}$$

$$\|\omega\|_2 \le 1 \tag{16}$$

Parameter τ denotes the marginal of value function between expert demonstration and other strategies. The maximal τ value is required by the method of maximizing marginal [19].

By formula (13) and (14), we can maximize the distance of value function between expert strategy and the optimal strategy by adjusting weights ω under the premise of that the optimal strategy is close enough to the expert strategy. In the end will be the only optimal solution.

3.2. The Solution Algorithm of Automatic Configuration Decisions Process for Virtual Machines in the Cloud

In order to gain reward function from experts demonstration, we assume that the expert strategy is π_E , thus a state sequence of expert demonstrations is obtained.

$$\left\{s_0^{(j)}, s_1^{(j)}, \ldots, s_t^{(j)}\right\}_{j=1}^{m} \tag{17}$$

We can get the expert character value vector:

$$\varphi(s) = \left(\varphi(s_0^{(j)}), \varphi(s_1^{(j)}), \ldots \varphi(s_t^{(j)})\right) \tag{18}$$

The variable $j (1 \le j \le m)$ denotes the times of expert demonstration, by equation (9), the expert character expectation value is available.

$$\mu(\pi_E) = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{\infty}\gamma^t\varphi(s_t^{(i)}) \tag{19}$$

Since then, we can solve the optimal strategy according to expert strategy π_E , the iterative algorithm is as follows:

Step1: Randomly select initial strategy $\pi^{(0)}$, plug it into equation (9), thus the character expectation of initial strategy $\mu(\pi^{(0)})$ is available， assume i=1;

Step2: repeat:

Step2.1: Solving equation (14), (15), (16), assuming $\tau = \|\mu(\pi) - \mu(\pi_E)\|_2$.

By 3.11: $\omega^{(i)} = \frac{\mu(\pi_E) - \mu(\pi^{(i)})}{\|\mu(\pi) - \mu(\pi_E)\|_2}$;

Step2.2: By equation (13),if $\tau^{(i)} \le \varepsilon$, end loop, else go to Step2.3;

Step2.3: Using reward function $R = \varphi(s) * (\omega^{(i)})^T$ of strategy $\pi^{(i)}$, by the method of value iteration or strategy iteration ,get the optimal strategy $\pi^{(i)}$;

Step2.4: Plugging strategy $\pi^{(i)}$ into equation (9), solve character expectation $\mu(\pi^{(i)})$;

Step2.5: set $i=i+1$, go back to Step2.1;

Step3: end;

4. Experiment and Analysis

4.1. Experiment Scheme

To test the technology of adaptive allocation for cluster based on inverse reinforcement learning, we design two kinds of test schemes:

1) program one:

Objective: to test whether the cluster can add or remove nodes automatically according to the variation of workload.

Approaches:

- Input three groups of different amplitude load, observe the adjusting number of nodes
- Input three groups of different frequency load, observe the adjusting number of nodes

2) program two:

Objective: to test the performance and efficiency of adaptive allocation for cluster.

Approaches:

- Input gradually increasing load, observe the response time of cluster

4.2. Test environment and Setting

We uses the virtualization software vSphere from VMware Co., through remote connection control of vSphere Client, we create a cluster of 38 VMs on the physical server in order to simulate the cloud environment. Among them 18 sets of VMs as server cluster, 18 sets of VMs as workload cluster, each VM’s configuration is as follows:

Table 2. VM’s Configuration

type / configuration	server	workload
OS	Ubuntu	Ubuntu
CPU	4 core	dual core
memory	8G	4G
disk	50G	50G
quantity	20	20

To simulate the workload, we write a program which can start 200 threads at the same time. Starting 18 workload VMs, each VM start 20 processes at the same time can simulate 72000 clients access the server at the same time. Set the initial number of running VMs for 3. The number can be increased to 18 later and can be reduced to only one VM. The formula that computes percentage of load input is as follows:

$$\lambda = \frac{load}{10 * time} \tag{20}$$

λ is the percentage of input load, the unit of which is *req/sec*. *load* is the number of clients. The standard period of input load is *T=360min*.

In order to generate allocation strategy through inverse reinforcement learning method, firstly, the author add or delete server nodes as an administrator according to workload change. Secondly, save all the system states as an "expert demonstration". Finally input all the states to system as commands.

4.3. Experiment Results and Analysis

4.3.1. Results of Program One :(1) Allocation results in situations of stable load:

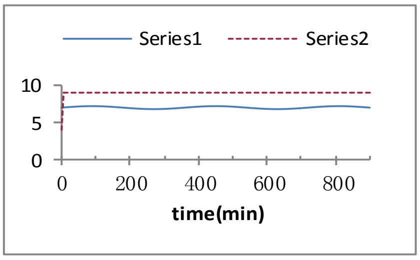
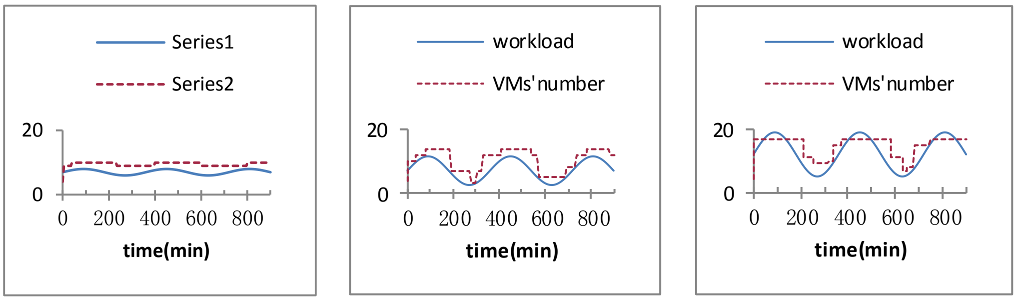


Figure 2. Allocation Behavior in Situations of Stable Load

From Figure 2, we can see the adjustment behavior when the system maintains 7.5. The initial number of VMs is 3. In order to satisfy the load request, it's increased to 8 when the workloads come. After that, the load almost tends to be stable. The number of clusters has maintained in 8 and there are no decisions to adjust.

(2)Allocation results in situations of different amplitude load:

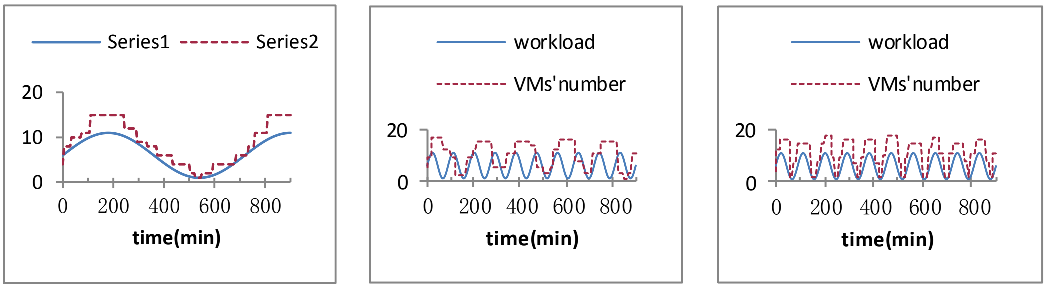


3(a). Workload $\lambda \in (5,10)$ 3(b). Workload $\lambda \in (2.5,12.5)$ 3(c). Workload $\lambda \in (5,20)$

Figure 3. Adjustment Behavior with Different Amplitude of Workload

Different adjustment behavior for cluster according to different amplitude of input load can be seen from the Figure 3. 3(a) shows that when the change of load amplitude is very small, the VMs cluster only output a decision-making behavior in order to reach the optimal reward function. Namely, the node's number is increased from 3 to 8. As there are no severe changes of load and reward value is relatively stable, the cluster almost not output more adjustment behavior. 3(b) and 3(c) show that the cluster output multi-step decision-making according to a dramatic change of load in amplitude. There is a positive correlation in frequency change between the VMs' number and the amplitude of workload .By 3(b), the VMs' number is adjusted from 3 to 10, then through multi-step adjustment, it maintains 14 for the sake of satisfying a larger amplitude of load .In Figure 3(c), the amplitude of workload is increased dramatically, the agent require more resources and the number is up to 17. When the load amplitude reaches bottom, the cluster also has an action of deleting nodes sharply.

(3)Allocation results in situations of different frequency load:



4 (a). Load frequency $f = \frac{1}{2T}$ 4 (b). Load frequency $f = \frac{2}{T}$ 4 (c). Load frequency $f = \frac{4}{T}$

Figure 4. Adjustment Behavior with Different Frequency of Workload

Figure 4 presents that different adjustment behavior for cluster according to different frequency of input load. According to Figure 4(a), for the workload which is half of the standard frequency, the agent made very detailed decisions to adjust. It increase or decrease nodes for many times and the number of adjustment is 1, 2 or 3 each time. So it can be concluded that the lower the frequency of the load, the finer granularity of adjustment. Figure 4(b) and 4(c) show that the agent still can adjust cluster for the faster input load. When load changes faster and faster, the number amplitude adding or removing nodes is bigger and bigger. As is shown in 4(c), the number adding or deleting nodes is 8 to 11 each time.

4.3.2. Results of Program Two:

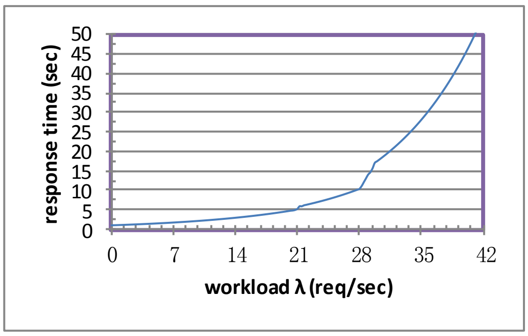


Figure 5. Response Time under Different Input Load

From curve above, we can see that the response time shows up a rising trend with the increase of the workload. It basically can be divided into four parts.

①$\lambda \in (0,14]$

The response time is less than 2s when workload maintains low level. The system can satisfy the user's request.

②$\lambda \in (14,21]$

The response time is between 2s and 5s when workload is gradually increasing.

③$\lambda \in (21,28]$

The load continues to increase, meanwhile, the response time is up to 5~10s accordingly.

④$\lambda > 28$

The response time rises sharply in the form of index is when the load is beyond normal levels.

In conclusion, when the workload is within 14 and there is no sharply increase, the response time of system is acceptable and basically meets the requirement of the performance due to factors such as process of decision-making and network transmission delay and so on. On the other hand, when workload is beyond a reasonable level, the response time is increasing, or even collapse.

4.3.3. Experiment Conclusion: The experiment one result in Section 4.3.1 shows that the adjustment behavior of system is not only related to the amplitude of the load at a certain moment, but also related to the change of load frequency. As for response time of system, we can see that there is gradual growth at different levels correspondingly as the workload changing in Section 4.3.2.

5. Conclusion

This paper puts forward a kind of automatic configuration management technology for virtual machines resources in the cloud. The main technical points are to introduce the reverse reinforcement learning into cloud resources management. The application system has the function of automatic learning. Second chapter solved the questions of modeling markov decision process. On this basis, the third chapter has solved the automatic configuration process for the virtual machine resources. The forth chapter shows the effectiveness of this technique by experiments, which can satisfy the automatic configuration management tasks for virtual machine and satisfy the user and the cloud provider's SLA requirements at the same time.

Acknowledgments

This work is supposed by the National Development and Reform Commission high-tech industrialization of Shanxi Province of China (No [2009]1365) and the Doctoral Scientific Research Foundation of Xi'an Polytechnic University (BS0752).

References

[1] T. Cihan, A.-N. Youssif and Akoglu, "Autonomic workload and resources management of cloud computing services", Proceedings - 2014 International Conference on Cloud and Autonomic Computing, ICCAC, **(2014)** January, pp. 101-110.

[2] A. Gandhi, M. Harchol-Balter, R. Raghunathan and M. A. Kozuch, "Autoscale: Dynamic, robust capacity management for multi-tier data centers", ACM Trans. Comput. Syst., vol. 30, no. 4, **(2012)**, pp. 1411–1426.

[3] D. Tsoumakos, I. Konstantinou, C. Boumpouka, S. Sioutas and N. Koziris, "Automated. Elastic Resource Provisioning for NoSQL Clusters Using TIRAMOLA", 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, **(2013)** May, pp. 34-41.

[4] J. Rao, X. Bu, K. Wang and C.-Z. Xu, "A Distributed Self-learning Approach for Elastic Provisioning of Virtualized Cloud Resources", Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE 19st Internatonal Symposium, **(2011)** July, pp. 45-54.

[5] A. K. Das, T. Adhikary, M. A. Razzaque and C. S. Hong, "An intelligent approach for virtual machine and QoS provisioning in cloud computing", 2013 International Conference on Information Networking (ICOIN), **(2013)** January, pp. 462- 467.
[6] C.-J. Huang, C.-T. Guan, H.-M. Chen and Y.-W. Wang, "An adaptive resource management scheme in cloud computing", Engineering Applications of Artificial Intelligence, vol. 26, no. 1, **(2013)**, pp. 382-389.
[7] R. D. C. Coutinho, L. M. A. Drummond and Y. Frota, "Optimizing virtual machine allocation for parallel scientific workflows in federated clouds", Future Generation Computer Systems, vol. 46, no. 5, **(2015)**, pp. 51-68.
[8] I. Konstantinou, E. Angelou, D. Tsoumakos, C. Boumpouka, N. Koziris and S. Sioutas, "TIRAMOLA: Elastic NoSQL Provisioning through a Cloud Management Platform", SIGMOD, **(2012)**, pp. 725–728.
[9] D. Hu-Yi, L. Gang, L. Han-Rong and C. Xin, "Nosql Evaluation in the Virtualization Environment", Journal of Air Force Early Warning Academy, vol. 27, no. 6, **(2013)**, pp. 449-451.
[10] V. Nikolov, S. Kachele, F. J. Hauck and D. Rautenbach, "CLOUDFARM: An Elastic Cloud Platform with Flexible and Adaptive Resource Management", IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), **(2014)** December, pp. 547- 553.
[11] X. Song, Q. Zhang and Y. Sekimoto, "Modeling and probabilistic resoning of population evacuation during large-scale disaster", Proceedings of the 19th ACM SIGKDD international congferrence on knowledge discovery and data mining. ACM, **(2013)**, pp. 1231-1239.
[12] L. Yaoyu, Z. Yifan, Y. Feng and J. Quan, "Inverse reinforcement learning based optimal schedule generation approach for carrier aircraft on flight deck", Journal Of National University Of Defense Technology, vol. 35, no. 4, **(2013)**, pp. 171-175.
[13] "Amazon EC2", Available: http://aws.amazon.com/ec2/.
[14] H. Jun, X. Wei, C. Luo and Y. Jiaxin, "Study and implementation of dynamic load balancing based on database cluster", Network and Communication, vol. 30, no. 2, **(2011)**, pp. 68-74.
[15] Shute, "F1: A distributed SQL database that scales. Proceedings of the VLDB Endowment", vol. 6, no. 11, **(2013)**, pp. 1068–1079.
[16] X. Xin, S. Dong, G. Yan-Qing and W. Kai, "Learning Control of Dynamical Systems based on Markov Decision Processes: Research Frontier and Outlook", Acta Automatica Sinica, vol. 38, no. 5, **(2012)**, pp. 673-687.
[17] J. Zhuo-Jun, Q. Hui, C. Shen-Yi and Z. Miao-Liang, "Survey of apprenticeship learning based on reward function learn ing", CAAI Transactions on Intelligent Systems, vol. 4, no. 3, **(2009)**, pp. 209-212.
[18] F. Gessert, F. Bücklers and N. Ritter, "ORESTES: a ScalableDatabase-as-a-Service Architecture for Low Latency", ICDE Workshops, **(2014)** March, pp. 215-222.
[19] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah and C. Hyser, "Renewable and Cooling Aware Workload Management for Sustainable Data Centers", ACM SIGMETRICS/Performance, **(2012)** June 11-15, London, UK.

Authors

Xue Tao, he is an Associate Professor in the Computer Science Dept at Xi'an Polytechnic University. He received his Ph.D. degree in Computer Science from the Xi'an Jiaotong University. He obtained his M.S. degree in Computer Science from the Northwestern University of China. His research interests include cloud computing, big data, and content-based networking.

Liu Long, Master in the Computer Science Dept at Xi' an Polytechnic University. Research interests include cloud computing, big data, and content-based networking.

