Research and Implemtation of A Software Online Testing Platform Model Based on Cloud Computing

Jing Chen
Shandong Provincial Key Laboratory of Computer Networks
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Jinan, China
College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, China
chenj@sdas.org

Chunxiao Wang
Shandong Provincial Key Laboratory of Computer Networks
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Jinan, China
wangchx@sdas.org

Fei Liu
Institute of Computing Technology
Chinese Academy of Science
Beijing, China
liufei01@ict.ac.cn

Yinglong Wang*
Shandong Provincial Key Laboratory of Computer Networks
Shandong Computer Science Center(National Supercomputer Center in Jinan)
Jinan, China
wangylscsc@126.com

***Abstract*—In a traditional process of software testing, an user have to buy servers and test tools to set up various test environment artificially, which causes high-cost, long test time and difficult test implements with limited resource. Cloud computing integrates a large number of physical resources into a pool and realizes the virtualization of various resources based on virtualization techniques. Users can achieve various virtual resources on demand from cloud platform without any investment on infrastructures. This offers a new solution to the problems in traditional software testing. So this paper proposes a multi-layer model of a software online testing platform based on cloud computing and implements the test platform. This test platform that include IaaS and SaaS platforms, self-help service and operation maintenance portals realizes automatic creation of test environments, remote use of test tools and online test services. Users can get various test services on demand only logging in to this platform at anytime and anywhere, which has the advantage of low test cost and high efficiency. In addition, this platform is also tested in terms of resource provision time and load variation, which shows the ability of this platform to satisfy users' resource demands.**

Keywords—cloud computing; software online testing platform; IaaS; SaaS; test service

I. INTRODUCTION

Software systems become more and more complex and large with their applications in different domains. Traditional techniques of software testing encounter with some problems , such as high test cost[1], long test time and difficult test implements with limited resources. Although some public welfare platforms provide various resources and test tools for users, these resources and test tools are limited to be used in the platforms. It is difficult to employ them to implement testing for most users, which cause waste of resources greatly. So it is the key problem how to build an effective test platform to benefit users' various demands based on these resources and test tools. Cloud computing integrates a large number of resources into a pool to simultaneously satisfy users' demands based on virtualization techniques[2]. Users can use various virtual resources on demand when they log in to cloud computing platform at anytime and anywhere, which not only saves users' cost on resources but also enhances the efficiency of their works. Therefore, cloud computing can bring a huge change for traditional process of software testing. Users need a software testing platform based on cloud computing to automatically build test environments, use test tools remotely and implement all kinds of test activities online.

Nowadays, some testing platforms or systems are set up to provide test services for plenty of users. For example, cloud testing is Parasoft's functional and load testing platform for cloud-based applications[3]. SOASTA's cloudtest is a load and perfomance testing tool to test websites and mobile applications quickly at any scale[4]. The testing platform, such as Testin from alibaba[5], Utest from tecent[6] and MTC from baidu[7], focus on testing on mobile applications. They mainly provide all kinds of mobile phones and services to debug or test mobile applications in terms of function, performance, security, compatibility and etc. However, these platforms cannot provide some basic test services with respect to traditional software, such as building test environments

*Corresponding Author
This work is supported in part by Shandong Provincial Natural Science Foundation (ZR2016FM41).

978-1-5386-1072-5/17 $31.00 © 2017 IEEE
DOI 10.1109/CBD.2017.23

automatically, using test tools at anywhere and executing test activities online. A cloud testing platform combining OpenStack and SAT was proposed to reduce test time and cost when massive concurrent test requests occur[8]. A software testing platform on cloud computing was proposed, which integrated IaaS and PaaS platforms[9]. Although these platforms can provide test resources, project management and code coverage testing, they cannot build various test environments, help consumers use test tools remotely and provide all kinds of online test services.

In order to alleviate these problems, this paper proposes a model to build a software online testing platform based on cloud computing. This platform can provide automatic creation of test environments, remote use of test tools and online test services, which can support a large number of users to carry out their test activities online. The key components of this platform are interpreted, such as IaaS platform, SaaS platform, self-help service portal and operation maintenance portal. Furthermore, the ability of this platform is also tested to verify whether it can satisfy users' demands in this paper.

II. A SOFTWARE ONLINE TESTING PLATFORM BASED ON CLOUD COMPUTING

A. Model of A Software Online Testing Platform

Test environments refer to various runtime environments of systems under test. For example, Tomcat7 and Mysql5.6 need to be deployed on a virtual machine in order to run a web site and test it. These runtime software can be integrated with an operation system into an image. Thus, a test environment can be constructed automatically when a virtual machine is created with such an image. Remote use of a test tool is also realized similarly, which integrates a test system into an image and use this image to create a VM with this test system. Users can use this test tool online by logging in to this VM. Some test services are developed in the form of SaaS applications. These SaaS applications support multi-tenants, isolate data of different tenants and can be integrated into a test platform. Therefore, we can build software online testing platform based on cloud computing to realize basic test activities in traditional software testing, such as automatic creation of test environments, remote use of test tools and online test services.

The model of software online testing platform based on cloud computing (Fig. 1) is a multi-layer model which integrates all kinds of physical resources into a pool, virtualize these resources, manage resources and services, and provides online test services for users. It includes the following layers.

- Physical layer: It provides physical compute, storage and network resources to support virtual resources provision and test services implementation.
- Virtual layer: It virtualizes physical resources based on KVM and components of OpenStack.
- Management layer: It manages physical and virtual resources of IaaS platform based on OpenStack, and manages service resources and service providers of SaaS by developing relevant components.
- Service layer: It provides services of test resource provision, remote use of test tools and online test services.

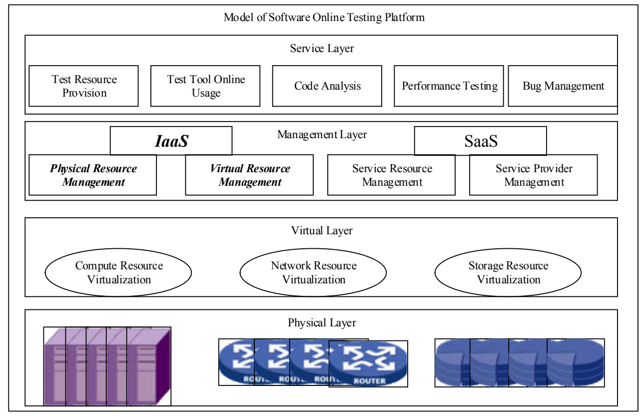

Fig. 1 Model of software online testing platform based on cloud computing

B. Construction of A Software Online Testing Platform

According to the above-mentioned model, we design a software online testing platform based on cloud computing (Fig. 2). This platform includes IaaS platform, SaaS platform, Self-help service portal for users, and Operation maintenance portal for administrators.

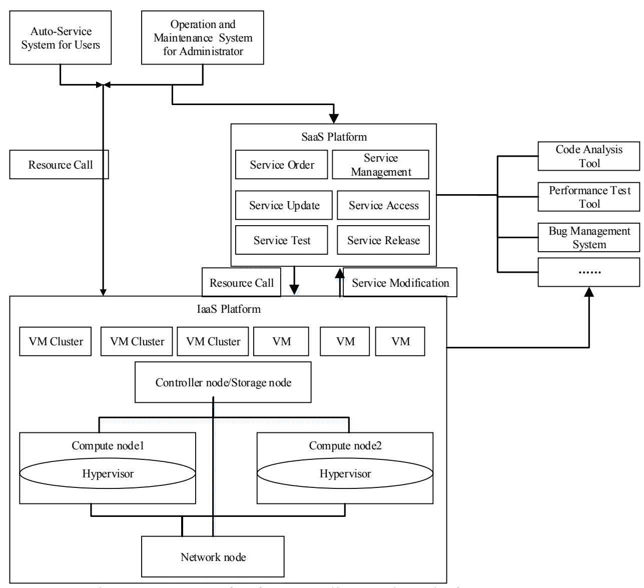

Fig. 2 Structure of software online testing platform

- IaaS platform: set up cloud infrastructure IaaS platform, which includes four nodes (i.e. physical servers) and each of them is deployed service components of OpenStack. The control node manages all physical nodes. The storage node is on the same node with control node which stores images and manages virtual disks. Two compute nodes create and manage virtual machines(VMs) through their hypervisors. A network node realizes communication between VMs from different compute nodes. A VM cluster can be constructed based on VMs and LBaaS service of OpenStack.

- SaaS platform: supports to access SaaS services from third-party Independent Software Vendors(ISV) according to access specifications of SaaS platform, and releases these services to software online testing platform for consumers' usage. These services can be deployed on either the servers of ISV or VMs of IaaS platform.
- Self-help service portal: provides services of test environment creation, remote use of test tools and software quality assurance service. These services are realized based on VMs created by calling APIs of IaaS platform.
- Operation maintenance portal: provides the functions of user management, business acceptance, resource management, service management, user rights management to help administrators operate and maintain the whole test platform.

III. IMPLEMENTATION OF IAAS PLATFORM

IaaS platform is built based on an open source cloud platform OpenStack [10,11], which includes hardware platform and software environment.

A. Hardware Platform

The underlying hardware platform in Fig. 2 employs four physical servers(64vCPU/128G/600G) with the operation system "Ubuntu 14.04". It adopts distributed control strategy. One center server is used as the control and storage node, which controls compute and network nodes by management network. It receives instructions from the upper portals and sends two compute nodes control messages to schedule resources and create VMs. Compute nodes can create more VMs by extending more compute nodes. These compute nodes can be divided into different zones, which can be selected when a VM is applied. A network node connects with two compute nodes by data network, which establishes communication between VMs.

IaaS platform use five VMs to deploy applications. One VM runs Tomcat server and one VM runs SQL server database for deploying Self-help service portal and Operation maintenance portal. Other three VMs are allocated to deploy test tools of software quality assurance service and relevant supporting components, such as code analysis tool, performance testing tool, bug management system, Jmeter and Sonar.

B. Software Environment

IaaS platform virtualizes physical resources by KVM techniques and manages various virtual resources based on OpenStack. OpenStack is an open source cloud management platform, which mainly schedules and manages resources based on service components, such as keystone, nova, glance, neutron, cinder, ceilometer and etc. Keystone realizes identify authentication and authorization, by which each component and system of OpenStack must be authenticated and authorized. Nova manages the whole lifecycle of virtual machines. It does not have the virtualization capability, but interacts with virtual machines by calling libvirt API. Glance mainly manages images. VMs cannot be created until an image is uploaded completely by calling glance API. Cinder is a storage management component, which can divide block storage to extend the hard disks of virtual machines. Neutron can be used to create private virtual networks and routers for each tenant, which can isolate network and data environments for different tenants in a multi-tenant cloud platform. We integrate four types of images, four test tools and three test environments in Table I. Certainly, users can also deploy various test environments by themselves based on VMs.

TABLE I. INTEGERATED IMAGES, TEST TOOLS AND TEST ENVIRONMENTS

Images	*Test Tools*	*Test Environments*
Windows 7	Appscan	IIS7/SQLserver2008
Centos 6.5/7 server	Loadrunner	Tomcat7/Mysql5.6
Ubuntu 12.04/14.04 Server	Diagnostics	Apache2.2/Mysql5.6
Windows 2008/2012 server	Fortify	

Test environments compared with general development environments are required to be independent of each other, namely better network and data isolation. This requests us to separate test environments from development environments, and create clean test environments without unnecessary software or data. It has a certain advantages of using virtual machines to create test environments based on cloud computing.

- Clean test environments can be created in batches by using images.
- Different users' data can be isolated well by setting different users and projects in OpenStack.
- Different users' network can be isolated well by creating private networks for each user

Thus, we use users, projects and private networks to isolate data and networks from different enterprises. If VMs between different enterprises need to communicate with each other, they can connect by route on virtual layer and GRE tunnel on physical layer.

In terms of resource scheduling strategies, there are some general methods, such as Random scheduling and Round-Robin scheduling. In addition, we propose a resource scheduling method based on load balance of testing platform. Suppose $l_{chj}, l_{mhj}, l_{dhj}, l_{nhj}$ separately denote the loads of CPU, memory, disk and network, and $w_{chj}, w_{mhj}, w_{dhj}, w_{nhj}$ separately denote their corresponding weight. Then, the comprehensive load of virtual machine v_h running on the physical machine p_j within the time period t is calculated by the following formula.

$$l_{hjt} = \frac{w_{chj} * l_{chj} + w_{mhj} * l_{mhj} + w_{dhj} * l_{dhj} + w_{nhj} * l_{nhj}}{w_{chj} + w_{mhj} + w_{dhj}}$$

The average load on the physical machine p_j can be obtained by summing the comprehensive loads of all virtual machines.

$$L_{jt} = \sum_{h=1}^{c} l_{hjt}$$

Suppose the new virtual machine $v_h^{'}$ causes the load of the physical machine p_j to increase $L'_{jt'}$ if it is allocated on this physical machine. The new load on the physical machine p_j is calculated by the following formula.

$$L_{ij} = \frac{w_{jt'} * L_{jt'} + w'_{jt} * L'_{jt'}}{w_{jt'} + w'_{jt}}$$

Thus, the standard deviation of the total load on the platform is calculated by the following formula under the mapping scheme S.

$$\delta_{ij} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(L_{ij} - \frac{1}{m}\sum_{j=1}^{m} L_{ij}\right)^2}$$

s.t. $\sum_{i=1}^{c+1} v_{ik} \le p_{jk}$, $k = 1,2,3,4$ denote CPU, memory, disk and network, respectively, that is, the total demands of virtual resources are less than the corresponding physical resource.

IV. IMPLEMENTATION OF SAAS PLATFORM

A. SaaS Access Environment

SaaS access environment involves service deployment environment and service configuration environment. Service deployment environment mainly carries ISV SaaS services or self-developing services on VMs created by calling APIs of OpenStack. The following API types are involved in Table II.

TABLE II. CALLED API TYPES BY SAAS SERVICES

API types	***Description***	***API types***	***Description***
Identify	Get token to use other services by identify API	Neutron	Create and manage network
Compute	Create and manage VMs by calling compute API and images.	Cinder	Create and manage cloud disk
Image	Upload and manage images	Ceilometer	Monitor and alarm VMs

VMs are created by calling compute API, and the real-time state and resource usage of VMs can be obtained by calling telemetry APIs. Certainly, more VMs (i.e. test execution machines) can also be added by continuously calling compute APIs when high load is imposed on initial VMs in the process of performance testing.

We develop configuration environment of SaaS service using C#, .NET. It includes the functions of ISV information management, service management, solution management, CMS management and etc. The information of enterprises, their SaaS services and integrated solutions can be put into the platform and can be managed well based on these functions.

B. Join Process of SaaS Services

The precondition of a SaaS service joining test platform includes the following items.

- It must adopt B/S structure. Users can get services by browsers.
- It must support multi-tenants and isolate applications and data for different tenants.
- It is chargeable and its pricing strategy should be sent to the administrator of this platform.

The process integrating a SaaS service into test platform is shown in Fig. 3. First, ISV information and its product(i.e. SaaS service) information are configured in the Operation maintenance portal. ISV can get the information of "ISV ID", "Key", "Log Account", "Product Number" and etc. Second, technical renovation of the product need to be implemented because the product must comply with the specification of test platform for users' access. It includes product modification and interface implementation. Product modification focuses on remove overlap of user interface or synchronize data between the platform and the product, such as user, role management, system logout, password modification as well as single sign-on. Interface implementations mainly involve service ordering, user authorization, roles, permissions, heartbeat keeping and etc. Third, the ISV product is tested in sandbox when the product is modified completely. Fourth, the product hit the shelf of the platform. Fifth, one or more products are integrated into a solution to be released. Sixth, a service content related with a solution is established, audited and released in CMS management. Thus, an user can order this service and use it when the order is checked. If a product expires, it will be off the shelf of the platform and will be deleted.

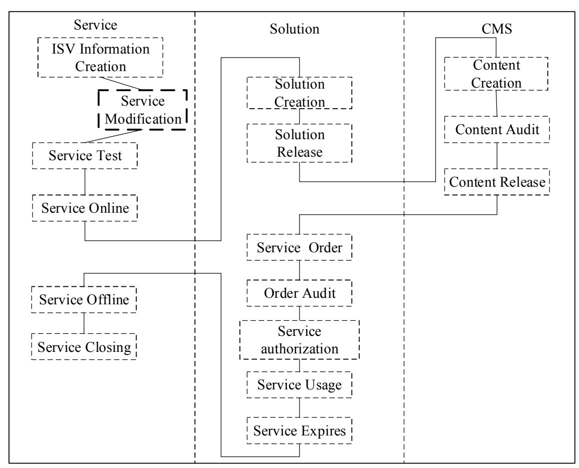

Fig. 3 Process integrating ISV services into test platform

C. Interface Implementation

Web service technology is applied in the main communication interfaces between test platform and ISV products. Simple Object Access Protocol(SOAP) is used to transform messages between them. Web services or Post page are adopted to receive data flexibly in interactive interfaces between them, and entity object or string (XML text) is used for the parameters and return values. They apply to .Net, Java and other heterogeneous platforms, such as PHP and ASP. Access interfaces from the platform and ISV products are presented in Table III in order to integrate SaaS services into SaaS platform.

TABLE III. ACCESS INTERFACES FROM PALTFORM AND ISV PRODUCT

Access Interfaces from Platform	*Access Interfaces from ISV Products*
SSO authentication interface	service interface for enterprise
SSO heartbeat keeping interface	service interface for enterprise staffs
Authentication interface of produc access	synchronous interface of role list(read)
Enterprisement information interface	synchronous interfaces of user role authorization (read/upfate)
enterprise staff interface	heartbeat keeping interface

V. SYSTEM IMPLEMENTATION AND VM SCALE TESTING OF SOFTWARE ONLINE TEST PLATFORM

A. System Imeplementation of Software Online Test Platform

We develop self-help service portal and operation maintenance portal using .NET, VS2010 and SQL server 2005. Fig. 4 and Fig. 5 show parts of their functions. Self-help service portal provides services of product ordering, order management, VM management, service management, enterprise information management and user management at the user level. It realizes automatic creation of VMs, remote use of test tools and online quality assurance service including code analysis, performance testing and bug management. Users can obtain these services at anytime and anywhere once they log in to software online test platform. They can deploy their systems on their own servers and use test services of self-help service portal to execute function testing or performance testing. Certainly, they can also deploy their systems on VMs and use test tools and test services to test these systems online through self-help service portal. The latter way can bring users more comprehensive testing and speed up testing process. Operation maintenance portal provides resource management, enterprise management, business reception, service management, ISV management, solution management, CMS management, user right management and etc. It can not only support resource and service provision, smooth operation and management of test platform, but also help administrators to improve the efficiency of operation maintenance.

B. VM Scale Testing of Test Platform

In order to support various test services, test platform need to provide enough VMs timely. So we perform stress testing on software online testing platform to show the capability of its resource provision. There are four physical nodes(servers) in this test platform. Each has 64Vcpu, 128G memory and 600G disk size. one of them is configured as controller/storage node, one is configured as network node and the other two are configured as compute nodes. The image “Centos6.5_64” and the flavor “standard1-1vCpu-2G-20G” are used in this test process. Because each VM uses 20GB disk from a compute node, the maximum number of VM requests is only up to 45 except disk use of images and VM files on two compute nodes.

Fig. 4 Self-help service portal

Fig. 5 Operation maintenance portal

We execute the stress testing through concurrently starting 10, 20, 30, 45 VMs, and show results in terms of provision time of an virtual machine and load of four physical nodes. Table IV demonstrates the provision time of an virtual machine under different numbers of VM requests. We can see that the provision time of an virtual machine normally is within the range [60s,120s] and it is gradually increasing with more and more VM requests. For example, the average provision time is 54.4s under 10 VM requests, while it changes into 61.45s, 85.7s, 101.2s under 20, 30, 45 VM requests, respectively.

TABLE IV. PROVISION TIME OF AN VIRTUAL MACHINE

Number of VMs	*Max time(s)*	*Min time(s)*	*Average time(s)*	*standard deviation*
10	62	47	54.4	5.35
20	73	48	61.45	7.46
30	96	57	85.7	12.6
45	119	72	101.2	14.1

Fig. 6 and 7 show CPU and memory utilizations of four physical nodes under the concurrent request of 45 VMs. Software online testing platform based on cloud computing can support concurrent startup of 45 virtual machines. The concurrent startup of virtual machines affects CPU utilization of the controller node. There is a peak pressure of CPU utilization up to 35% when 45 VMs just start concurrently. And CPU utilization almost retains 10% in the subsequent startup process of VMs. Memory of the controller node hardly changes with the use rate about 28.5%. This concurrent request also has little influence on network node in terms of CPU and memory utilizations, while it impact resource utilizations of two compute nodes. It can be seen that two compute nodes afford almost the same CPU load due to using load balance. Their CPU utilizations both present a dynamic change by ascending quickly up to 17.5%, retaining a high load in a period, and then descending to retain a value. Memory utilization of compute1 node changes from 5.8%(7GB) to 12%(14GB), while memory utilization of compute2 node changes from 8.2%(9GB) to 14.1%(16GB). The factual consumption of memory amounts to 14GB for two compute nodes. However, 45 virtual machines should occupy memory of 90GB. The reason may be that thin provisioning technology improves memory utilization. From the above analysis, we can see that this platform is limited in disk capacity, and other physical resources are enough to ensure good running of VMs if disk is satisfied.

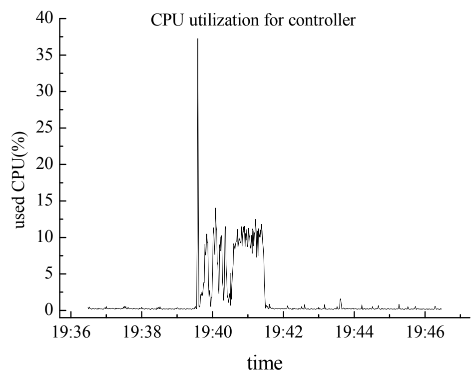

Fig. 6-1 CPU utilization for controller node

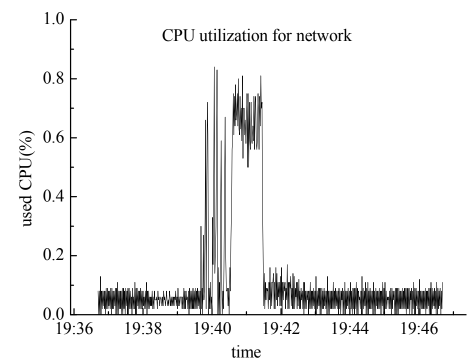

Fig. 6-2 CPU utilization for network node

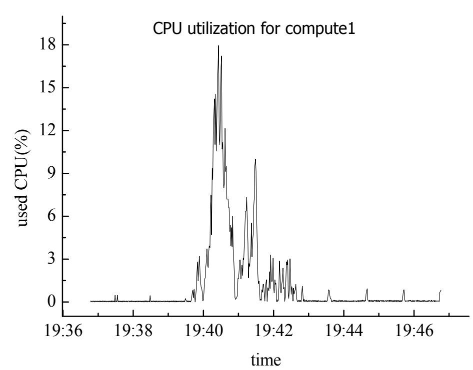

Fig. 6-3 CPU utilization for compute1 node

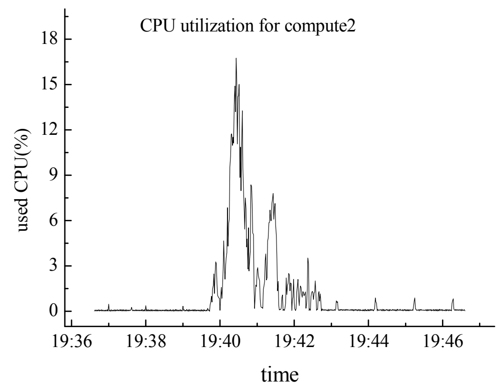

Fig. 6-4 CPU utilization for compute2 node

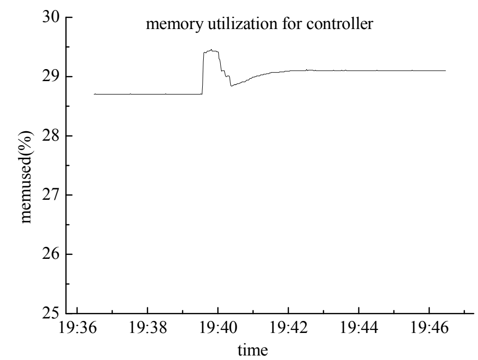

Fig. 7-1 memory utilization for controller node

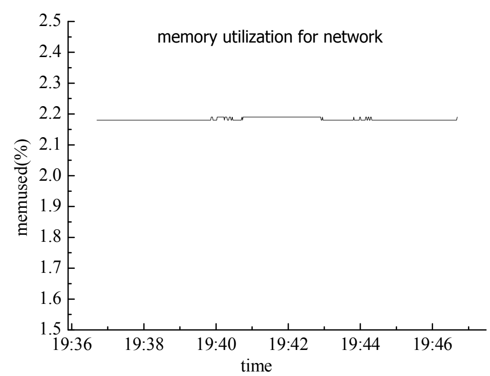

Fig. 7-2 memory utilization for network node

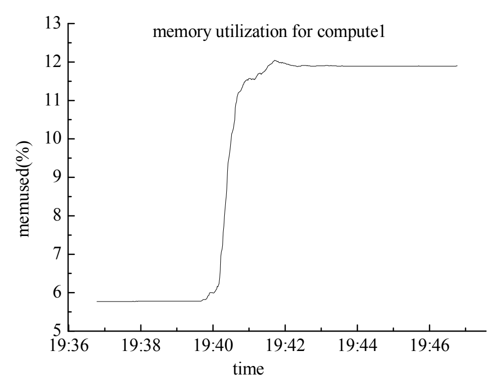

Fig. 7-3 memory utilization for compute1 node

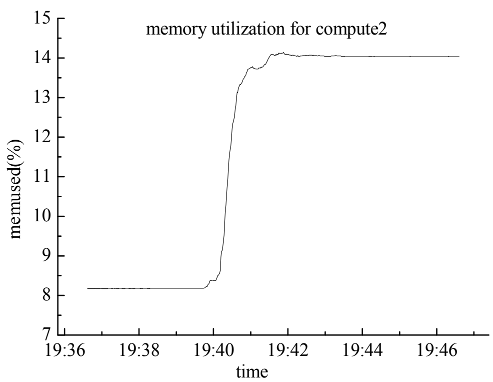

Fig. 7-4 memory utilization for compute2 node

VI. CONCLUSION

In order to solve the problems of high-cost, long test time and difficult test implements in traditional software testing, we proposes a multi-layer model of a software online testing platform based on cloud computing in this paper. Based on this model, we build a real software online testing platform, which integrates IaaS platform, SaaS platform, self-help service portal for users and operation maintenance portal for administrators. This platform provides automatic creation of VMs, remote use of test tools and online quality assurance service to satisfy users' basic testing demands. We also test the VM scale that this platform can support. The result demonstrates that the average provision time of an virtual machine normally increases with more VM requests involved. Compute and controller node are affected by the concurrent startup of VMs while network node is hardly impacted by it. Based on the model of software online testing platform or its partial techniques, we have implemented multiple actual projects for customers which have validated the applicability and stability of this model.

ACKNOWLEDGMENT

We thanks for Wenxiang He and Bin Zhang for their suggestions on this paper.

REFERENCES

[1] Q. Li, D. L. Ke and X. L. Wang, "Brief Survey on Cloud Testing," Application Research of Computers, Vol. 29, No. 12, pp. 4401-4406, 2012.

[2] T. Cheng, "Key Technology of Cloud Computing and Current Research," Journal of Langfang Teachers college(Natural Science Edition), Vol. 13, No. 2, pp. 41-45, 2013.

[3] Cloud testing. Functional and Load Testing Platform. https://www.parasoft.com/capability/cloud-testing/

[4] SOASTA. The Cloud Test Platform. https://www.soasta.com/load-testing/.

[5] Testin. The Cloud Test Platform. http://www.testin.cn/.

[6] Utest. The Cloud Test Platform. http://utest.qq.com/.

[7] Baidu MTC. The Cloud Test Platform. http://mtc.baidu.com/

[8] X. P. Ding, H. Hou, S. He, Z. G. Chen and D. H. Guo, "Performance Study of Cloud Testing Platform Based on OpenStack," Software, Vol. 36, No. 1, pp. 6-10, 2015.

[9] L. Cao, Y. Jiang, C. M. Gan, Y. C. Zhang and G. Q. Chen, "Construction of Software Testing Platform on Cloud Computing," XIANDAI TUSHU QINGBAO JISHU, No. 11, pp. 34-39, 2012.

[10] F.L. Fan and C. L. Yao, "The system implementation of complex product data warehouse based on Open Stack cloud computing platform technology," 2013 International Conference on Information and Communication Technology for education, Singapore, pp. 121-128, 2013.

[11] L. M. Wu, H. B. Wang and S. D. Cheng, "Research on Virtual Machine Launching Mechanism of OpenStack," The Journal of New Industrialization, Vol. 2, No. 8, pp. 28-32, 2012.