The Online Education Platform using Proxmox and noVNC technology based on Laravel Framework

Lei Chen
School of Computer Science
Communication University of China
Beijing, China
chenlei163mail@163.com

Wei Huang, Aina Sui
School of Computer Science
Communication University of China
Beijing, China
huangwei1983@gmail.com, aina_sui@163.com

Deqin Chen
School of Information Engineering
Communication University of China
Beijing, China
chendeq1015@163.com

Chengsheng Sun
China Electronics Technology Network Information
Security Co., ltd.
Chengdu, China
sunchengsheng@dingtalk.com

***Abstract*—Currently, Massive Open Online Courses (MOOC) is having significant impact on E-learning by providing free campus curriculum for millions of viewers over the world. Most of MOOC mainly include literal tutorial, video course and easy quiz, but lack of an identical and easily operational experiment environment with durable data. It is crucial for participants to learn complicated courses involved practical programming and multi-host network interaction. In this paper, we focus on the construction of an E-learning platform based on Laravel framework implementing the customized virtual experimental environment, which in accordance with teachers' requirements, by using Proxmox virtualization technology and VNC client noVNC based on HTML5. In this way, every experiment can be configured with multiple virtual internet hosts wherein user operation and data is saved durably so as to conduct more complicated experimental operation next time**

Keywords—E-learning platform; Laravel; Proxmox; noVNC

I. INTRODUCTION

With the rapid development of internet technology in recent years, online education is gradually prevalent. Massive Open Online Course(MOOC) have been fashionable around the world since 2012 and a great number of world-famous elite schools have already established the online education platforms which mainly aim at courses taught in institutions of higher learning [1]. The teaching modes mainly include reading materials, video courses, examinations and some online programming environments, such as Codecademy [2] and iMOOC, etc, but its functions are only limited to the simple online programming environment and it cannot implement some complicated experiments, such as operating system, distributed computation and information security, etc. Considering the online education platform providing the complicated experimental environment, the online virtual security laboratory [3] proposes to create virtual machines containing the complicated experimental environment for learners. There are similar studies such as MOOC based openHPI [4], practical experiences with InstantLab [5]. Online experiments are operated by using users' terminals to access virtual machines, so as to complete experiments with complex environment. Moreover, it has no need to take lots of time to construct the complicated local environment. However, the experimental platform structure is extremely complex and has some limitation, so it is difficult to satisfy the demand that multiple virtual machines with different systems realize interconnection and finish the experiment. Additionally, configuration of every virtual machine may not be founded for users' private customization. In this paper, we focus on how to design and realize the online education platform that can configure as required. With the consideration of development efficiency and maintainability of the platform, we adopt the mainstream PHP development framework Laravel. Main functions of the online education platform include experiment module, course module and quiz module. The virtual machine management platform based on Proxmox is applied for the sake of making users unpack virtual machines to test relevant virtual machines. Furthermore, it is necessary to use noVNC client technology based on HTML5 and realize the connection between the platform and Proxmox by combining with Nginx reverse proxy technology. At last, the complicated experimental tasks can be fulfilled by embedding remote desktop into the front-end iFrame, controlling and operating remote virtual machines.

II. ARCHITECTURE OF PLATFORM BASED ON LAREVEL

In order to develop the online education platform rapidly and effectively and make the platform system holding the high reusability, stability and safety, we adopt the Laravel framework which can standardize the development process and deal with non-business logic relation automatically to make developers concentrate on the realization of business logic [6]. The compositions of platform framework based on the Laravel framework are shown in Fig. 1, including front-end page

978-1-5090-5507-4/17/$31.00 ©2017 IEEE
ICIS 2017, May 24-26, 2017, Wuhan, China

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18,2024 at 03:44:20 UTC from IEEE Xplore. Restrictions apply.

module, access control module, course teaching module, quiz module, experiment module and database.

The construction of the front-end page will make the most of the flexibility of blade module engine and combine with the front-end framework jQuery and Bootstrap to design interactive dynamic effects. Our platform uses multi-role management, including teachers, students and administrators. For each role, the limit control will use the built-in authentication facilities of Laravel. In addition to using authentication controllers via registry when logging in, it will also add some methods relating to authentication in controllers for each function.

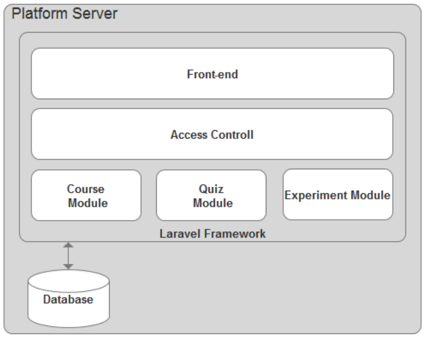


Fig.1. Platform Server Architectures

The main structure of permission control is shown in Fig. 2. As to different users, such as teachers and students, when they access to the website by using the corresponding identity information, they will obtain the authentication controller to judge the identity. After judging the legal identity and logging successfully, functional controllers for each module can execute different code logic by judging roles so as to allow roles to use the available functions under the role permission and return to the corresponding front-end page.

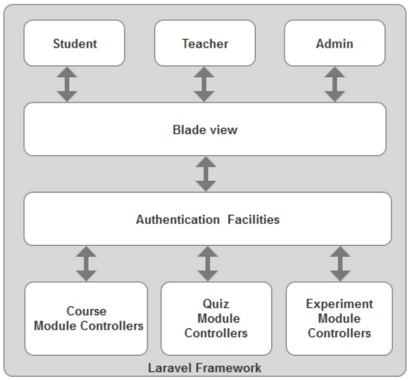

Student
Teacher
Admin
Blade view
Authentication Facilities
Course Module Controllers
Quiz Module Controllers
Experiment Module Controllers
Laravel Framework

Fig.2. Access Control of multi-role

Rules fabricated by the platform are designed according to the primary MOOC platform functions analysis [9]. It is convenient for students to look up tutorials, use relevant experimental virtual machines established by teachers and accept online testing. Meanwhile, teachers can increase, delete, modify and check tutorials and test questions and also configure the quantity of relevant experimental virtual machines, CPU memory, network connection mode, and mirror of operating system, etc. In addition, teachers can supervise students' learning situations, true-false answers, as well as submit an experimental report to give comments on students' experimental reports and give feedback to students in time. Besides using all functions of students and teachers, administrators also can forbid or ban posting for all users, adding, editing and deleting users, and conducting configuration operation with the higher permission for virtual machines.

In the platform, our course modules mainly support rich text reading materials and video course materials. For teachers authorized, they can add information of test questions manually based on jQuery plugins. Forms of test questions can include single choice, multiple choices, gap filling and short answer questions. Besides, we can upload test questions in the form of word based on the specified standards. Our platform can transfer the format automatically and generate online quiz. The experiment module requires teachers to upload experimental guidance which is to guide students to do online experiment. Teachers can assign virtual machines to let students accomplish a test. Results and feedback information are compiled after students upload experimental reports. Teachers can score them and judge whether they are passed. Here, the technical realization of experimental functions is emphasized as follows.

III. Online Experimental Implementation Procedure

Main technologies of online experimental functions in this paper are shown below:

A. *Proxmox Solutions*

Whereas the platform should allow users to create lots of virtual machines and manage it, we use the open-sourcing virtual solution, Proxmox, which can provide Web interface (interface web) and visual interface. It can manage the user virtual machine, storage pool (pool storage), node and so on [10]. All of them are based on Corosync Cluster Engine [11]. It maximizes the limit of data capacity of RAM, so it is enough for Proxmox to establish thousands of configuration documents of virtual machines. By using Proxmox, we can not only manage the uploaded mirror documents of the system, but also start, close, destroy and migrate virtual machines, even create new virtual machines from the existing templates. When encountering with software bug or hardware collapse, Proxmox has strong fault-tolerant ability. Furthermore, by combing with server API provided by Proxmox website [12], requests can be launched on the Laravel directly to configure and operate virtual machines in cloud environment.

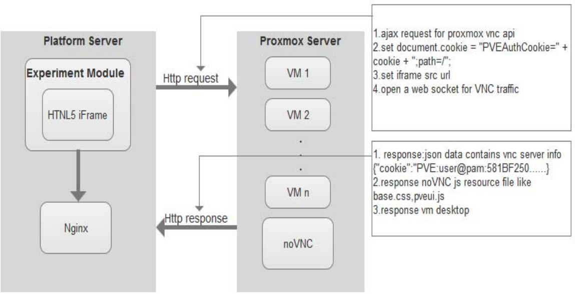
Proxmox solutions can satisfy configuration demands of the stronger virtual machines and more complicated network configuration, so that robustness, safety and concurrency of the platform will increase with it.

B. *VNC Technology*

In order to make users operate virtual machines in our platform remotely, we adopt noVNC solution which offers Web Sockets agency of VNC server and JavaScript, so that

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18,2024 at 03:44:20 UTC from IEEE Xplore. Restrictions apply.

users can operate host interface via cloud environment in client remotely [13]. With such a desktop cloud computing framework, it can reduce unnecessary constructive time of systematic environment [14]. Users can conduct an online experiment on any place with browsers and use noVNC technology to return the process of remote virtual machine interface on our platform. In the experiment module, when users click start, it will acquire necessary ticket for server communication by requesting to access VNC interface in Proxmox. The return contents of ticket involving in JSON string containing the cookie. Then, the cookie value of the current user interface is set up, as shown in step three in Fig.3.



1.ajax request for proxmox vnc api
2.set document.cookie = "PVEAuthCookie=" + cookie + ";path=/";
3.set iframe src url
4.open a web socket for VNC traffic

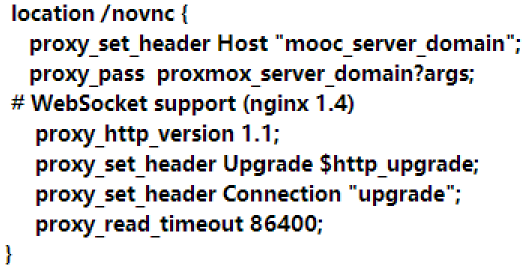
1. response:json data contains vnc server info {"cookie":"PVE:user@pam:581BF250......}
2.response noVNC js resource file like base.css,pveui.js
3.response vm desktop

Fig.3. Server Interaction Process

Source address of iFrame element in front-end interface for users should be constructed like this:

"https://username:password@domain/novnc/?console=kvm&novnc=1&node=proxmox&resize=scale&vmid = vmid".

Then it has access to the assigned interface of virtual machines for users (virtual machines of Proxmox use the only identification of vmid). The format of the address requires for user's name and password of Proxmox, so as to confirm Proxmox servers with permission and to complete the above-mentioned process to access remote interface in a browser. It can be observed from Fig.3 that there is an Nginx module with the main functions of solving server resources out of the front-end JavaScript code cross-domain access to platform server, so as to return relevant resources of VNC server to our platform.



```
location /novnc {
  proxy_set_header Host "mooc_server_domain";
  proxy_pass  proxmox_server_domain?args;
# WebSocket support (nginx 1.4)
   proxy_http_version 1.1;
   proxy_set_header Upgrade $http_upgrade;
   proxy_set_header Connection "upgrade";
   proxy_read_timeout 86400;
}
```

Fig.4. Nginx Configuration Segments

As shown in Fig .4, with the simple Nginx settings, when we access the cross-domain website in our education platform domain, it will give request to other host automatically. By setting up the attribute Upgrade and Connection, it allows to establish a channel between a client and a back-end server which supports for Web Socket proposal. In this way it may acquire essential resources including pictures and JavaScript code in the VNC server.

C. *Experimemt Environment Customization*

Through our platform, teachers and administrators can create multiple virtual machines for every experiment with customized configuration to meet the needs of the course environment. As shown in Figure 5, the user can configure the virtual machine image system according to the experimental requirements. We can provide various versions of the system ISO for Windows, Linux. In addition, users can set the hard disk size, the number of cores and sockets of CPU, memory size. Besides, network mode settings have three options like namely bridge mode, NAT mode, no network adapter.

In order to improve the user experience in our E-learning platform, we designed the friendly interactive effect of the front page more attractive, simplified some optional configuration .After creating the virtual machine, teachers or admin can scan the list of virtual machines on the experiment management page, where they can edit the created host to adapt to the new requirements and changes.

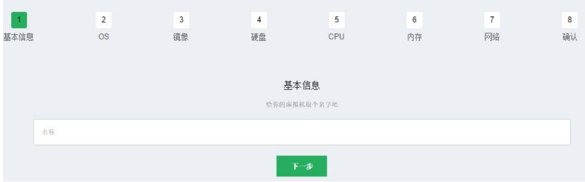


Fig.5. Virtual Machine Creation

IV. Experiment Results And Analysis

In this section, the characteristics and environment under which online course is designed are described. It is assumed that an experiment with regard to the course of information security need a simulation experiment environment consists of three virtual machines, each separately representing for the provider (CentOS 5.1), attacker (Ubuntu 16.04) and victim (Windows 7). Aaccordingly, when teacher or admin upload an experiment course, firstly, they should select appropriate virtual machine template images relating to these three systems from system list only if the experiment task has been related to OS image.

A. *Set Up Experiment Environment*

As shown in Fig.6, after preparation work is done, all virtual machines on the experiment requirements list would be clone and then start up through Proxmox cloud, in addition, the virtual machine should be restarted it has been started before.

The first time starting procedure of virtual machine might be very slow for reasons of allocation of enough resources, including memory and CPU resources. Finally, our platform will generate three usable virtual machines automatically to meet needs of experiment for the course.

On account of a great many concurrent user requests meanwhile or quite frequently operation within a short time interval, it may lead to failure of clone or start up for virtual machines required by experiment environment. In light of this situation, we took special measure, instant recovery function, to recover all the virtual machines cloned or started up

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18,2024 at 03:44:20 UTC from IEEE Xplore. Restrictions apply.

unsuccessfully in experimental host list. On the basis of database referred to virtual machine ID and template configuration info, to resume executing a program of cloning virtual machines from template and starting up virtual hosts until all virtual machines could be usable. To be specific, virtual template information such as name, cores and socket number and operating system images would be packaged into form of array so as to be merged into HTTP requests interacting with Proxmox server. In addition, network mode, for instance, NAT mode and bridge networking model should be taken into account. After that, users will continue to take practical tasks in virtual experimental environment providing multiple host users can switch back and forth just like different roles in real environment.

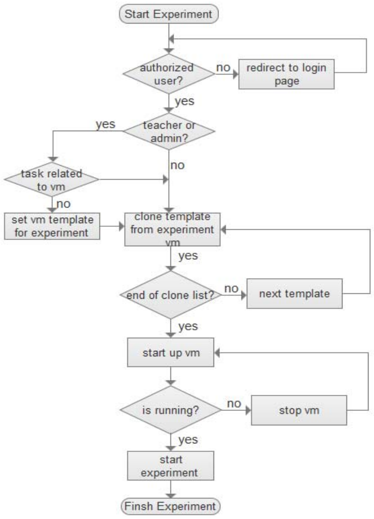
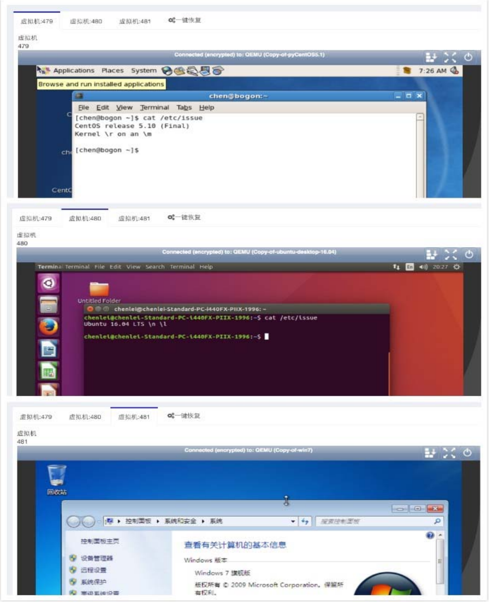


Fig.6. VM Construction Process

B. Experiment Result



```
[chen@bogon ~]$ cat /etc/issue
CentOS release 5.10 (Final)
Kernel \r on an \m

[chen@bogon ~]$
```

```
chenlei@chenlei-Standard-PC-i440FX-PIIX-1996:~$ cat /etc/issue
Ubuntu 16.04 LTS \n \l

chenlei@chenlei-Standard-PC-i440FX-PIIX-1996:~$
```

查看有关计算机的基本信息

Windows 版本

Windows 7 旗舰版

版权所有 © 2009 Microsoft Corporation。保留所有权利。

Fig.7. Online Experiment Interface

When users access to our platform to do the experiment, the corresponding virtual machine template will be cloned according to the template. The virtual machine is private to every user. Data can be saved permanently. Virtual machines can access to each other. A user enters into the front-end page in the experiment and it is shown in Fig.7. It can enter into the corresponding interface of different virtual machines by switching label pages, so as to do a complicated experimental operation by using interconnection of multiple operating systems on our platform.

V. Performance Evaluation

In terms of our experiment module in our platform that is built on the basis of the remote virtual machine related to online practical tasks, one of the evaluation indexes for the user experience is the response time.

Response time of the website has an important influence on user experience of the online education platform [12] and impacted by several factors like network conditions, software interfaces and VM provisioning techniques [16]. As a result of our platform deployment and operation mainly distributed among the inner campus network, we temporarily neglect access speed test in multiple regions and simply test several core functional points in the campus network, including homepage, log in and log out page, course page, quizzes page, experiment page, operation of virtual machines like create, start and stop, screen refresh, etc. In process of this analysis, we adopted brief JavaScript code embed in every test section consists of dynamic or static web pages. As shown in Fig.8, when log in website with user and password derived from HTML Form, the following function code will be executed asynchronously for recording response time in millisecond in log page, meanwhile, test method of response time for other pages is similar to this.

```
function loginPage()
{
    var sendTime = getClickButtonTime();
    Do While receive ajax response
        set CSRF headers;
        set request method;
        set url;
    End While
    If StatusCode == 200
        return getCurrentTime() - sendTime;
    Else
        return -1;
    End If
}
```

Fig.8. Response Time Test Function

The histogram of response time for every part is depicted in Fig.8. It can be observed that main modules should have good response speed within a second that users fell imperceptible [16], nevertheless, creation, start up and stop of virtual machines, and resource distribution in cloud environment wasted a lot of time.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18,2024 at 03:44:20 UTC from IEEE Xplore. Restrictions apply.

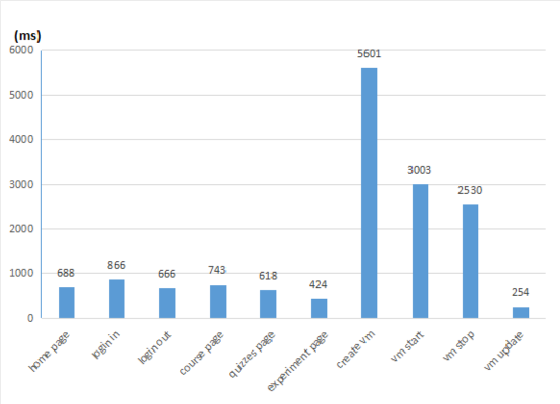


Fig.9. Response Time

Concurrency means the number of requests that the system can handle simultaneously, thus, it reflects the load performance of a website. In the case of a large amount of users access our E-learning platform at same time, it is essential to measure concurrency of the platform .By using a performance testing tool in the Pylot, one open source free tool for load testing web applications. Our testing case design on URL of tutorial homepage, setting up 500 concurrent requests and implementing it for 10sec. As shown in Fig.9, with the increase of concurrency, the quantity of response resources also will be increased. At present, our education platform can cope with thousands of users' concurrent requests and distribute resources rationally.

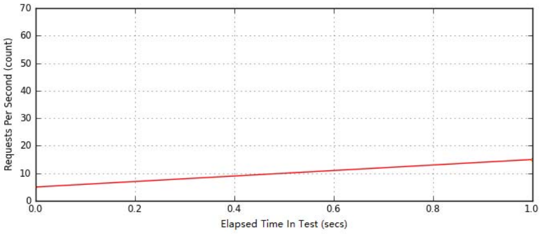


Fig.10. Throughout

VI. Conclusions And Futhure Research

The purpose of proposing the online education platform aims at improving the defects of the existing MOOC platform, including weak flexibility in online experimental module to satisfy for practice task. The platform make users just need a browser, but have no need to configure the experimental environment of any hardware to realize online experimental courses of multi-host interconnection. With our experiment module, teachers can set up detailed experimental environment by themselves, while students can access and operate multiple virtual machines under cloud environment according to experiment settings. These virtual machines can have different operating system, different hardware and network configurations. Moreover, data of every virtual machine can be saved and privatized permanently. The platform development is based on the powerful framework of Laravel, so as to improve development efficiency of a team and enhance maintainability and safety. By using Proxmox to manage virtual machines in cloud environment, noVNC can control remote interface. At last, we also test lag time and concurrency of the website. It basically can satisfy demands of many users, but it also has some problems to be improved, such as longer foundation time of virtual machines and delayed operation, etc., resulting in a great influence on user experience. Next, we intend to enhance the improved scheme for virtual machines' response speed, deploy the campus network platform to the internet, and analyze and test it by aiming at more users.

References

[1] L. Pappano, "The Year of the MOOC," The New York Times, vol. 2, no. 12, 2012.

[2] R. Ruth, "Codecademy is teaching the world programming, for free," Startup Hook [online] http://startuphook.com/employment-2/codecademy-teaching-world-programming-free/498, 2015.

[3] C. Willems and C. Meinel, "Tele-Lab IT-Security: an Architecture for an online virtual IT Security Lab," iJOE, vol. 4, pp. 31-37, 2008.

[4] C. Willems, J. Jasper and C. Meinel, "Introducing hands-on experience to a Massive Open Online Course on openHPI," in Proceedings of IEEE International Conference on Teaching, Assessment and Learning for Engineering, pp. 307-313, 2013.

[5] C. Neuhaus, F. Feinbube and A. Polze. "A platform for interactive software experiments in massive open online courses," Journal of Integrated Design and Process Science, vol. 18, pp. 69-87, 2014.

[6] H. YU, "Design and implementation of web based on Laravel framework," Atl. Press, no. Iccet 2014, pp. 301-304, 2015.

[7] J. Duckett, "JavaScript and JQuery: Interactive Front-End Web Development," John Wiley 1 edition, 2014.

[8] P. Hall, "The bootstrap and Edgeworth expansion," Springer Science & Business Media, 2013.

[9] J. I. Kim, "A Study on the K-MOOC Platform Standardization Measures," International Journal of Software Engineering and Its Applications, vol. 9, pp. 221-236, 2015.

[10] A. Kovári and P. Dukan, "KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE," in 2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, pp. 335-339, 2012.

[11] S. C. Dake, C. Caulfield and A. Beekhof, "The corosync cluster engine. In Linux Symposium" , vol. 85, 2008.

[12] K. A. Elias and A. Ahmad, "The Design of The Laboratory Exercise Using Virtualization Technology For A System Administration Course." in Proceedings of International Conference on Information Technology & Society, 2015

[13] D. Mulfari, A. Celesti, M. Villari and A. Puliafito, "Using virtualization and noVNC to support assistive technology in cloud computing," Network Cloud Computing and Applications (NCCA) on 2014 IEEE 3rd Symposium, pp. 125-132, 2014.

[14] C. Neuhaus, F. Feinbube and A. Polze, "A platform for interactive software experiments in massive open online courses," Journal of Integrated Design and Process Science, vol. 18, pp. 69-87, 2014.

[15] R. Bao, "Test and Analysis of Website Response Time of e-Learning Institutions in China" Open Education Research, 2006.

[16] B. Taylor, Y. Abe and A. Dey, "Virtual Machines for Remote Computing:Measuring the User Experience," Carnegie Mellon University , 2015

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18,2024 at 03:44:20 UTC from IEEE Xplore. Restrictions apply.