Application of Programming Experiment Platform Based on Docker Container

Shan ZHOU¹, Longxin LIN¹,⁎, Zhenyu HE¹ and Yan SHU²

¹College of Information Science and Technology Jinan University, Guangdong Guangzhou, China
²School of Economy and Management, Guangzhou University of Traditional Chinese Medicine, Guangzhou, China

⁎Corresponding author

Keywords: Programming basic, Virtualization, Docker, Container Technology.

Abstract. The programming foundation course which based on Python has become an important basic course for different disciplines in universities. In experimental teaching, the main difficulty is the environment configuration and rapid deployment of Python. Virtualization technology is a common method to solve this problem. However, there are some problems, such as low resource utilization, insufficient support for concurrent, slow start, and so on. So, a new method for constructing the teaching experiment platform of the course using Docker container technology is proposed. This method can quickly realize the dynamic allocation, deployment and management of the running environment image according to the needs of students. Experimental results show that this method has obvious advantages in terms of resource utilization, concurrent users, response time, construction cost and so on, compared with virtual machine technology.

Introduction

Recently, with the rapid development of big data, data science, Internet of things, cloud computing, artificial intelligence, etc., Python-based programming courses have become increasingly popular. At present, this course has become a public compulsory course for non-computer majors and an elective course for computer majors in universities across the country. Python as a simple and flexible “glue” language, after nearly 30 years of development, through its strong third-party extension capabilities, has formed a strong “computing ecosystem” across all disciplines and fields, especially in the above areas. The power of Python depends on its rich third-party extension library, which can be used by students of different disciplines and majors to solve specific professional problems through Python programming. Thus, also give a Python program design course introduces new challenges: pure Python installation configuration is simple, and all sorts of closely combining with professional third-party extensions library configuration installation, configuration, and deployment of teaching experimental environment is relatively complex and diverse, in large scale, more professional, more class’s and grade’s teaching demand, the rapid deployment of teaching experimental platform in the experimental environment, resource utilization, supported by the maximum number of concurrent users put forward higher requirements.

In response to the above challenges, at present, many universities use virtualization technologies such as Vmware and KVM to build teaching laboratories, and use cloud desktop to provide students with a variety of experimental environments, which to some extent simplifies the deployment, maintenance and management of servers. However, in actual operation, this method has the following disadvantages:

(1) Single user’s resource occupation is too high, resulting in high construction cost. In the scheme of virtualization technology, the whole operating system needs to be virtualized to run multiple virtual machine instances on one physical server. In the specific experimental teaching activities, it is necessary to run an independent virtual machine instance for each user (such as students), and allocate independent memory, storage and network resources for it. No matter how many resources the experimental task consumes, it is necessary to configure fixed computing resources and storage
resources for each virtual machine instance, which undoubtedly leads to a huge waste of resources and greatly increases the construction cost.

(2) Insufficient support for the number of concurrent users. For the above reasons, in a specific experimental activity, it is necessary to configure fixed virtual machine resources for each user, resulting in the maximum number of concurrent users supported to be fixed configuration completely according to the actual situation of software and hardware platform resources, which cannot be adapted and adjusted by the actual experimental task dynamic state, resulting in the low utilization rate of system resources in fact. In the actual virtual machine solutions, the utilization ratio of computing resources and memory resources of the actual physical system is often less than 50%, but more virtual machines can no longer be allocated to meet the new user needs.

(3) The virtual machine starts slowly and the maintenance management is complex. Based on the traditional virtual experimental platform, in the use process, if the teaching needs change, the entire operating system image needs to be updated, and the stability and compatibility of the system needs to be tested again, resulting in cumbersome management and maintenance process of the platform. In the experimental teaching, different courses need to switch different virtual machine images. If the class is large (for example, more than 100 people) and the time between courses is short (10-15 minutes off between classes), the restart of more than 100 virtual machine images usually takes several minutes, and the switching speed is very slow.

In view of the above situation, this paper proposes a new method of using docker container technology to build the teaching experiment platform of program design course, which can quickly realize the dynamic allocation, deployment and management of running environment image according to the needs of students. Based on this method, a teaching experiment platform for students of different majors is constructed. This platform provides each student with an independent experiment environment based on docker container dynamically. When the experiment is finished, the container resources can be released quickly. Experimental results show that compared with virtual machine technology, this method has obvious advantages in resource utilization, concurrent users, response time, construction cost and so on.

Docker Introduction

Container Technology

Container technology is an operating system level virtual technology. By sharing the host operating system, multiple independent system environments can be created in the host. No special system library is needed to provide the required functional modules for each system environment [1]. Container is to package the application and its dependent environment into a mirror image, providing users with an independent running environment, including file system, running space, etc., which is not limited to any programming language, framework or web container. It can simplify the deployment and testing process of the program running environment, quickly reuse in any operating system, and is lighter and more efficient than the virtual machine [2].

Docker

At present, there are many container management tools in the market, and docker is the most recognized one. Docker is a lightweight container engine launched by dotcloud in 2013 [3]. Docker uses client / server architecture and remote API to manage and create docker containers. Docker daemon is a daemons running on the server, which is responsible for accepting and responding to the client’s requests, including build, push, pull and other commands to create and start containers [4].

Key components of docker include image warehouse, image and container [5]. Image warehouse is a collection of images. Image is a read-only template and an independent hierarchical file system. Each image is composed of a series of layers. When the user needs to configure the image, docker will add or upgrade a new layer in the image layer, and finally encapsulate it into a standard, isolated, lightweight environment. The configuration will not have any impact on the original image.
is a running instance created according to the image. When docker runs the container, all operations on the container will be written to the file system of the container, that is, the writable layer of the image. Therefore, you can use the same image to start multiple containers, which are isolated from each other and do not affect each other [6].

Docker uses dockerfile to create a docker image. Dockerfile is a file instruction set. Each instruction corresponds to a Linux command, which is used to build the main components and each layer of the container [7]. The docker engine is responsible for the translation of instructions. When the image needs to be modified due to the change of requirements, just add or modify the instructions in the corresponding dockerfile [8].

Making Docker Image

There are many images on docker hub for users to pull, but if you want to get images that fully meet your professional needs, you’d better make them by yourself. Docker officially provides two ways to create images:

The first way is to use the basic image to generate a container. Users modify the container according to their own needs, and the modified content will generate a new image layer on top of the basic image. Then use the docker commit command to commit the container to the image warehouse.

The second way is to build the dockerfile file according to your own needs, and then build the image through the docker build command. The dockerfile syntax specification, it can clearly show how the image is generated. It is easy to learn, and highly repeatable. Therefore, this paper mainly uses this method when making the image.

Design and Implementation of Experimental Platform

Design of Experimental Platform

The teaching experiment platform designed by docker technology is mainly divided into the following three steps:

(1) Build a hardware environment for the experimental platform. According to the number of concurrent users of the platform in the same period, we chose two servers with better performance and build the hardware environment of the platform. Then install the Ubuntu Linux operating system on the server, obtain the latest version of docker installation package, start the docker background service, and build the docker container environment under the Ubuntu system.

(2) Generate the experiment image library. According to the needs of students of various majors, we use the dockerfile instruction to create a new image. By modifying the command in the instruction set, we complete the installation of the required application software package and the loading of environment variables. Then we use the “docker build” command to build the image and generate the application image library.

(3) Using image to carry out the experiment. When carrying out the experiment teaching, students choose the image with the software package needed by the specialty to generate and start the container, and they deploy, operate and test the experiment program in the container to solve their practical problems.

Implementation of Experimental Platform

According to the above design scheme, the implementation steps of the platform are as follows:

(1) Deploy the basic image. We choose computers with better hardware performance as servers, deploy basic images, and install docker running environment on Ubuntu 16.04.

(2) Using the dockerfile method to make the image of the running environment of the experimental program required by the basic course of program design for each specialty. The key steps include: install Python, JDK, SSH, mysql, anaconda, tensorflow and other common programs and corresponding library files, and configure the environment.

(3) Configure the system environment variables, Java home, home, path.
(4) Configure SSH Remote access user name and password.
(5) Use the “docker build” command to generate a series of application image sets, which contain the compilation environment required by each specialty.
(6) Upload the image to the official image warehouse or image warehouse of other platforms, so that students can download the image on the private computer.

According to the above design scheme, the implementation steps of the platform are as follows:
(1) Experimenter: the experimenter is responsible for collecting and summarizing the needs of students of various specialties, installing various applications and configuring the environment. Make, manage, and maintain a collection of mirrors on the server.
(2) Students: during the experiment, students send docker instructions to the platform. After receiving the instructions, the docker daemon process sends the corresponding instructions to the image warehouse and pulls the basic image required by students to the server. According to the image, the server generates the corresponding container and returns it to the student side. The student uses the container to carry out the experiment. After the operation, the container can be saved and sealed again to build a new image and upload it to the image library for other students who need to download.

![Platform activity flow chart.](image)

**Experiment and Analysis**
In order to verify the effect and performance of the experimental platform, we choose the same hardware equipment: two dawn servers with high performance, equipped with Intel e5-2620v46 core/12 thread CPU, 16GB DDR4 memory, 2TB hard disk. We install Ubuntu 16.04 Linux operating
system for the server and build different experimental platforms respectively. One is the traditional
virtualization experimental platform. The server deploys VMware virtualization software.

We set up another docker teaching platform. All devices are connected through two 48 port Gigabit
switches. Students access the experimental platform through the thin terminal. After the environment
is built, simulate the teaching scene, start VM virtual machine and docker instance step by step, and
carry out the same experiment of Python course. The performance of the two platforms in memory
occupation and response time is shown in Figure 2 and Figure 3.

![Figure 2. Comparison of memory consumption.](image1)

Through the analysis, we know that with the increase of the number of experimental groups, the
memory consumption and response time of the two experimental platforms show a similar linear
growth trend.

On the traditional virtualization platform, the server allocates 2G memory for each virtual machine.
When about 15 virtual machines carry out experiments at the same time, the memory of the server is
almost exhausted, and the response speed of the server becomes slower and slower with the
consumption of memory until it cannot respond.

Compared with the virtualization platform, each docker instance launched by the docker based
experimental platform consumes about 200m memory of the server. When 35 groups of experiments
are started, the memory consumption is about 20%. After testing, the server memory capacity can start
up about 160 groups of experiments at most. Experiments show that the container platform can carry
more experimental groups than the virtualization platform with the same server configuration. The
new platform has the following advantages:

1. In terms of management, the platform helps managers to manage the laboratory more
efficiently. The platform has the characteristics of incremental distribution and modification update.
When there is an error in the image, you can modify it partially through the instruction without
affecting other containers. When teaching needs change, the whole operating system does not need to
be reinstalled.

2. In terms of operating efficiency, docker is virtualization at the operating system level. It directly
reuses the operating system of the server. The platform only takes a few seconds to start. Compared
with traditional virtual machines, it is a lighter and faster virtualization.

3. In terms of resource utilization, a server can have thousands of containers in the same time. The
containers only occupy resources when running, which has higher resource utilization than
virtualization.

4. In terms of construction cost, all the configurations of the platform are completed on the server,
and there is no need to install too much software on the client. The requirements for client
configuration are not high, which saves the investment in hardware cost.

5. In terms of teaching efficiency, the container can run and expand on multiple platforms, and
students can transfer the applications to different operating systems for reuse, without the need to
redploy complex compiling environment, which improves the learning efficiency.
Summary

Experimental teaching is essential to the teaching of programming courses. This paper builds docker experimental platform, which provides a rich and efficient compiling environment for Python experimental teaching. Students can access the server anytime and anywhere, run docker container to carry out experiments. On the one hand, the platform can reduce the cost of laboratory construction and maintenance under the same laboratory scale; on the other hand, it can reduce the difficulty of students’ experimental learning and improve the efficiency of experimental teaching under the same teaching objectives. At present, the platform mainly provides services for Python experimental teaching. In the future, we will continue to study the characteristics of other courses, expand the functions of the platform, and deploy more experimental environments for non-computer majors.

Acknowledgement

This paper is partially supported by the following funding: “Reform and Practice of College Computer Fundamentals Using Data as the Mainline”, Guangdong Provincial Higher Education Teaching Research and Reform Project, 2017; The teaching research and practice of computer courses for non-computer majors in the fragmented era”, Higher Education Teaching Reform Project of Guangdong Province, 2016; “A study on the innovation of the course system of trade major in medical colleges and Universities under the background of Guangdong, Hong Kong and Macao From the perspective of empowerment theory” (2019-152). The corresponding author is Longxin Lin.

References

[1] XiaoLan Xie, Peng Wang, Qi Wang. The Performance Analysis of Docker and Rkt Based on Kubernetes[C]. 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017).2017:2137-2141.

[2] DiLiu, Li, LiBin Zhao. The research and implementation of cloud computing platform based on Docker [C]. 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). 2014:475–478.

[3] Nitin Naik, Building A Virtual System of Systems Using Docker Swarm in Multiple Clouds[C]. IEEE International Symposium on Systems Engineering (ISSE). 2016:1-3.

[4] Rajdeep Dua, Vaibhav Kohli, Sriram Patil, Swapnil Patil. Performance Analysis of Union and CoW File Systems with Docker[C]. International Conference on Computing, Analytics and Security Trends (CAST). 2016:550-555.

[5] Xie B, Sun G, Ma G Docker based overlay network performance evaluation in large scale streaming system[C]. Advanced Information Management Communicates, Electronic and Automation Control Conference (IMCEC), 2016IEEE.IEEE.2016: 366-369.

[6] Tyler H. Brandom S., et al., Slacker: Fast Distribution with Lazy Docker Containers, 14th USENIX Conference on File and Storage Technologies (FAST’16), 2016.

[7] N. Naik, “Applying computational intelligence for enhancing the dependability of multi-cloud systems using Docker Swarm,” in IEEE Symposium Series on Computational Intelligence (SSCI), 2016.

[8] Day, S.: Docker Registry V2—A New Model for Image Distribution. In: Docker Con 2015. San Francisco (2015).