

[Original Paper]

Development of Server System Construction Training System

Mitsuyoshi KITAMURA[†], Takumi YAMAGUCHI[†], Toshikazu TAKESHITA[†], and
Takahito TOMOTO[†]

[†] Graduate School of Engineering, Tokyo Polytechnic University

Abstract

This paper reports on the development of a server system construction training system (SSCTS) designed to educate prospective server engineers. Since server system construction, operation, and management issues are vital knowledge for server engineers, they must comprehensively learn the details of server system monitoring, construction, troubleshooting, etc. In order to achieve this, a learning system using multiple real servers is normally required for each learner (user). This inevitably means that there will be numerous issues that must be overcome in terms of cost, operations, and maintenance, particularly when preparing an environment where multiple users can learn simultaneously. The SSCTS makes this possible by using server virtualization technology and virtual network computing.

The system is based around the four main learning menus of “Monitoring”, “Server construction”, “System construction”, and “Troubleshooting”, which were prepared with an emphasis on server system construction, operation, and management. In each menu, users acquire knowledge while interacting with the SSCTS to engage in examination and construction activities, and each user can access and use up to 25 virtual servers simultaneously. Under normal circumstances, a system with multiple virtual servers that are presented as one unit will require long time periods for system porting and other expansion activities, and there will be concerns regarding the possibility of human errors caused by manual operations. To address these problems, we developed and introduced an automatic porting system for the SSCTS and then verified the system’s effectiveness via the following experiment.

Taking the construction of a failover cluster system as an example, two groups, Group A (which used the SSCTS) and Group B (which used a real server system consisting of multiple servers), participated both in the experiment and in examinations presented before and after system construction to determine their comprehension improvements. Then, the two groups exchanged systems, and the system construction process was repeated, after which a questionnaire regarding operability was given to both groups. The results of the pretest and posttest examinations showed definite and similar levels of knowledge improvements for both groups, thus indicating that as a training system for server engineers, the SSCTS is superior to a real server system in terms of cost and power consumption.

1. Introduction

The novel coronavirus pandemic has emerged as one of the greatest threats to modern society and has caused numerous problems in various fields, including economics, medical care, and education. One of the basic countermeasures to prevent the spread of this virus is to avoid human-to-human contact. Therefore, the roles of information systems required for teleworking, online education, and other online activities are extremely important. However, Japan is now faced with the inadequacy of its teleworking environment and a lack of infrastructure necessary for online education. Because of this, the nation is making efforts to accelerate the introduction of information technology (IT) and to expand and strengthen the availability of Internet services [1]. However, it has also been pointed out that Japan is facing a shortage of IT human resources required to develop and operate these systems [2].

With those problems in mind, various studies aimed at reducing the loads on IT-related system operators have been conducted [3]-[9]. In addition, research on learning systems related to IT-based education systems and server systems, including networks, has advanced [10]-[25]. However, many of the studies teach limited functions, and there are no systems for comprehensively imparting all of the required knowledge. Basic knowledge such as operating system (OS) installation, system monitoring, server construction, server system construction, and troubleshooting is particularly important for server engineers. Since prospective server engineers must comprehensively learn these topics, an education system using multiple real servers was previously required for each learner (user), and numerous issues related to cost, operations, and maintenance were inevitably encountered when preparing an environment where multiple users could be educated simultaneously.

[Original Paper] Received 13 August 2021, revised 11 October 2021, accepted 27 October 2021.

© Information Systems Society of Japan

© 情報システム学会

In response to these issues, we have developed a server system construction training system (SSCTS) for the purpose of training prospective server engineers. This system is based on CentOS (Linux), which is often adopted as the OS for servers and employs Kernel-based Virtual Machine (KVM) server virtualization software and virtual network computing (VNC) to allow multiple users to simultaneously interact with the SSCTS, which runs on one computer. Because the SSCTS focuses on server system construction, operation, and management, it supports the following four main learning menus: “Monitoring”, “Server construction”, “System construction”, and “Troubleshooting”. In each menu, “active learning” is performed while users actually engage in examination and construction activities, and each user can access and use 25 virtual servers simultaneously. However, since the high number of virtual servers in the SSCTS would normally require a lengthy construction time when adding a user, we developed and introduced an automatic porting system for the SSCTS and then verified the effectiveness of the system via the experiment described below.

Taking the construction of a failover cluster system (FOC) as an example, this experiment examined user comprehension levels in reference to the presented learning materials. Specifically, participants were divided into two groups: Group A (which used the SSCTS) and Group B (which used a real server system consisting of multiple servers). Tests were presented to each group before and after system construction to examine user comprehension improvements, after which the groups exchanged systems, and the process was repeated. After completing the second set of system constructions, a questionnaire regarding operability was presented to both groups. The results of this experiment show the superiority of the SSCTS as a training system for server engineers and for improving the knowledge required to construct a server system.

The remainder of the present paper is organized as follows. Related research is described in Section 2. Next, the SSCTS configuration, its functions, and the automatic porting system required to add users are shown in Section 3. Section 4 then describes the outline and functions of the learning menu, which is focused on server system construction, operation, and management. Next, Section 5 describes the FOC construction experiment mentioned above, including details on the experimental method, the use of comparative examinations for determining comprehension levels imparted by both the SSCTS and the real server system, and an analysis of the experimental results. Finally, we provide our conclusions in Section 6.

2. Related research

Various studies have been conducted or are now underway aimed at reducing the loads on IT-related system operators. For example, in the field of troubleshooting with a focus on data centers, an integrated design that considers cloud service problems such as a single-link failure, service failures on data center networks, and data center network placement was discussed [3]. Separately, hardware problems and service abnormalities were evaluated in relation to server security and network resources of a virtual data center [4], and a high-availability virtual infrastructure management framework that considers the problem rate of data center devices was reported [5].

Since servers are the fundamental units of systems providing services in data centers, methods that facilitate rapid failure recovery efforts by improving data redundancy before a serious server failure or network error occurs are essential. This can be accomplished by detecting impending failures in servers that construct high-availability distributed clusters along with minor network abnormalities that can affect cluster operation [6]; the development of a multiple-server backup system that can back up the functions of several real servers using virtual servers started from one real server [7]; proposing a high-availability server system and a peer-to-peer method server management system that does not require a specific management server [8]; and the introduction of a power-saving, high-availability server system with two types of power-saving server system configurations that consider the server system operating environment [9].

In addition, research on learning systems related to IT-based educational and server systems, including networks, is advancing. For example, in the field of e-learning, a method that improves the efficiency of learning data analysis by arranging learning content from the cloud to the network edge was proposed [10]; as an education management system, the design and implementation of an application system based on a cloud computing platform were discussed [11]; a Web server educational system that adopts Docker container technology and a developed education management and learning system was proposed [12]; the development of a model to analyze large-scale data collected in education using mobile terminals and the realization of

feedback that fits educational needs was reported [13]; and the construction of a virtual learning environment that supports interactive virtual reality to provide a better learning experience was discussed [14].

Furthermore, in the field of basic learning, a programming learning system that focuses on essential components such as the algorithms used in programming education was proposed [15]; an e-learning system that supports programming education was discussed [16]; the construction of a learning environment that stimulates and evaluates programming-related thinking in operations during exercises was proposed [17]; the construction of a Linux operating system (OS) installation exercise environment using virtual machines was discussed [18]; and an exercise system in which multiple learners can simultaneously construct one network was advanced [19].

As for security, on the subject of hijacking Internet of Things (IoT) devices, network security education that works by exposing trainees to actual denial of service (DoS) attacks was proposed [20]; the development of a low-cost offensive and defensive network security exercise system that considers the attacker's perspective, as well as the defender's perspective, was discussed [21]; and server intrusion exercises using a virtual server with preset vulnerabilities for the purpose of raising security awareness were reported [22].

In the field of server management, the development of a system that records the entered commands and contents of edited configuration files during server management exercises was proposed [23]; the planning and operation of a server configuration training program, including vulnerability evaluations and the remediation process, were discussed [24]; and the construction of a training environment with an advisory function for system administrators and a learning model proposal for server administrators presented as a browser application were reported [25].

However, many of the abovementioned studies only teach limited functions, and there are no systems for comprehensively imparting all of the required knowledge. In basic IT service management (ITSM) training courses, students comprehensively study system construction basics, security management basics, ITSM basics, IT basics, and other product knowledge [26]. Our SSCTS covers all of those course items except product knowledge. Moreover, in IT education, it is necessary to acquire multiple information technologies such as cyber security, information networks, architecture design, and system integration, which means a curriculum for training IT engineers with a wide range of knowledge is also necessary, as has been shown [27]. Our SSCTS can comprehensively teach security, system construction, operation, management, and server virtualization technology to prospective server engineers.

3. SSCTS outline

3.1. System configuration for SSCTS

Figure 1 shows the system configuration for SSCTS. As mentioned above, the SSCTS is based on CentOS (Linux), which is often used as an OS for servers, and uses KVM and VNC in order to make it possible for multiple users to train on the system at the same time. The SSCTS, which is mounted on a single computer equipped with Domain Name System (DNS) and VNC server functions, provides individual training systems configured for each user. Each user uses a VNC client to access the SSCTS, and connections with the client are divided by Network Address Translation (NAT) for each user login. This allows each user to execute the training system in an independent environment. For example, when logging in User 1, NAT 1 is used for communication. Additionally, if the SSCTS is used via the Internet, it can be realized by installing a Virtual Private Network (VPN) server.

In the SSCTS training system, operation, server construction, and system construction manuals are provided to facilitate user operability improvements. There are also two clients: Client 1, which is used for accessing each server, and Client 2, which is used to examine the connection settings with those servers. There are also seven general servers (example servers) to provide normal functions, which are prepared as examples for demonstrating server settings. Additionally, there are five initial training servers (on which only the OS is installed) for use in actually constructing the server system, and 11 servers for use in configuring the sample server system.

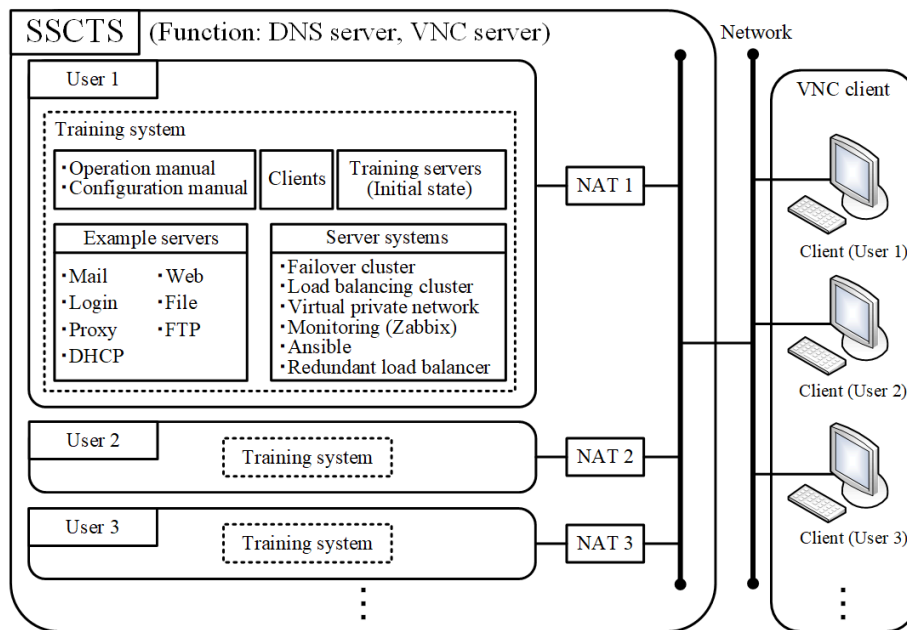
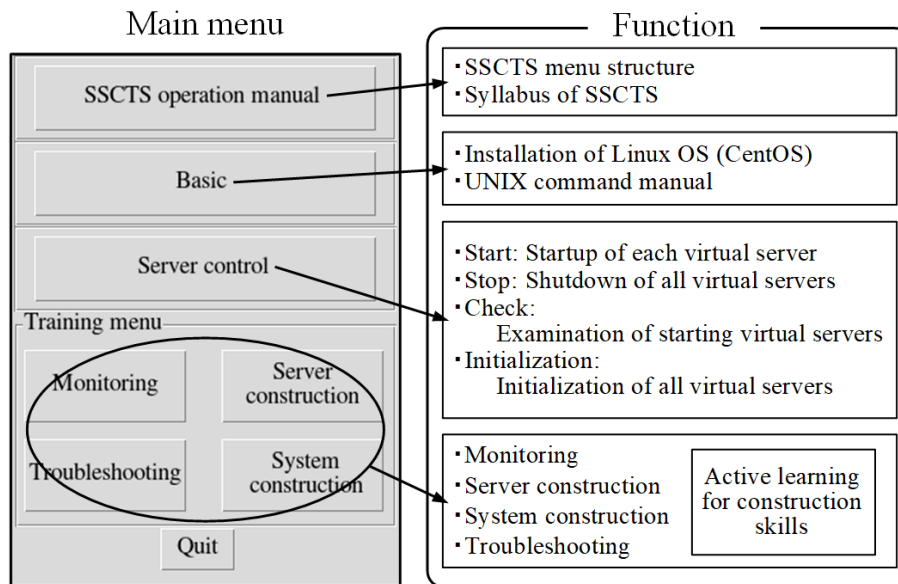


Fig. 1: System construction for SSCTS.

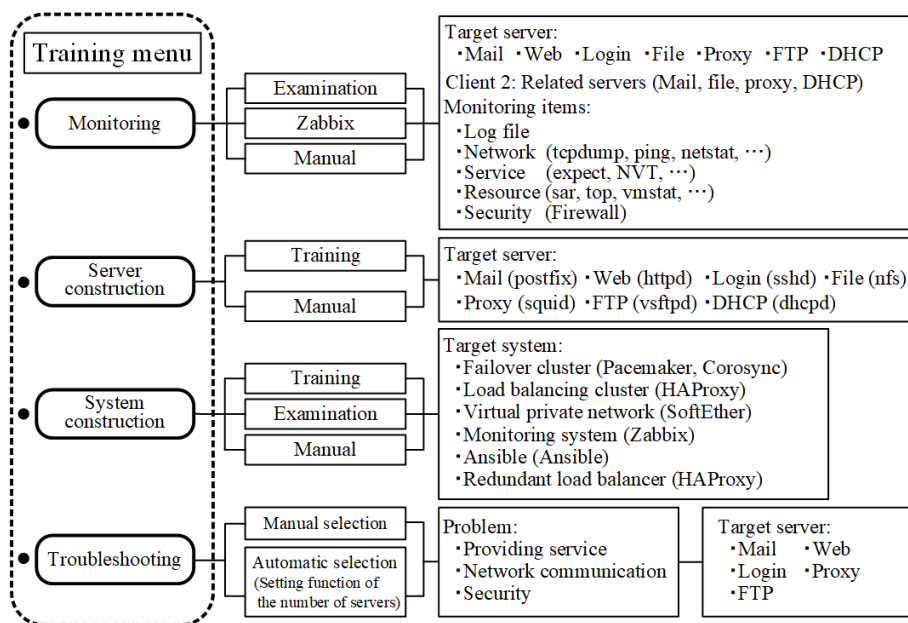
To simplify operations, the SSCTS adopts a menu selection method that is executed by selecting a button according to the desired content. Figure 2(a) shows the main menu. The structures of each menu and the learning syllabus contents can be examined in the “SSCTS operation manual”. In “Basic”, the fundamental lessons and exercises related to installing a CentOS and learning UNIX commands can be performed, while in “Server control”, controls related to starting and stopping the virtual server used in the training menu, an examination covering the server start status, and the initial settings for the servers are available.

Details of the training menu are shown in Fig. 2(b). When considering server system construction, operation, and management, it is important that prospective server engineers acquire server knowledge on monitoring technologies, server and server system construction technologies, and fault countermeasure methods. Therefore, the SSCTS has four main menus: “Monitoring”, “Server construction”, “System construction”, and “Troubleshooting”. In the Monitoring menu, an environment for learning the command operations required for server monitoring is provided. Actions that can be performed while referring to the manual include examining the server log, monitoring network packets using the UNIX “tcpdump” command, examining the service offer and resource usage statuses, and examining the security setting status.

The settings and statuses of Client 2, which handles tasks related to mail, file, proxy, and Dynamic Host Configuration Protocol (DHCP) servers, can also be examined. In addition, server monitoring using Zabbix can be performed. In the Server Construction menu, an environment for learning server construction is prepared, and server construction exercises using the training server can be performed while referring to the manual. After construction, the server operation is examined via the method learned in the monitoring menu. In the System Construction menu, an environment for learning the construction process is prepared, and actual construction is performed using the training server. In addition, users can examine the operating status of the constructed system while referring to the manual. Here, users can use the practical skills acquired when studying via the Monitoring and Server Construction menus. In the Troubleshooting menu, the “Manual selection” option allows the faulty server and fault type to be selected manually, while the “Automatic selection” option allows them to be selected randomly, thus making it possible for users to engage in realistic troubleshooting exercises.



(a) Main menu



(b) Training menu

Fig. 2: Menu configuration and functions.

3.2. System construction

The SSCTS uses a menu method in order to improve operability. Because this system was developed on Linux, the UNIX “Tcl/Tk” toolkit was used for developing the menu display system. Figure 3 shows the menu display program for server selection, in which users can set the title, window size, font, button name, and command to be executed when it is selected. For example, when the “Mail” button is selected, the program mail.sh (UNIX C shell procedure) that starts the mail server is executed. The “&” in the program indicates execution in background processing. Therefore, because multiple of these buttons can be selected, multiple servers can be started at the same time. When the “Quit” button is selected, the window is closed, and the program is terminated.

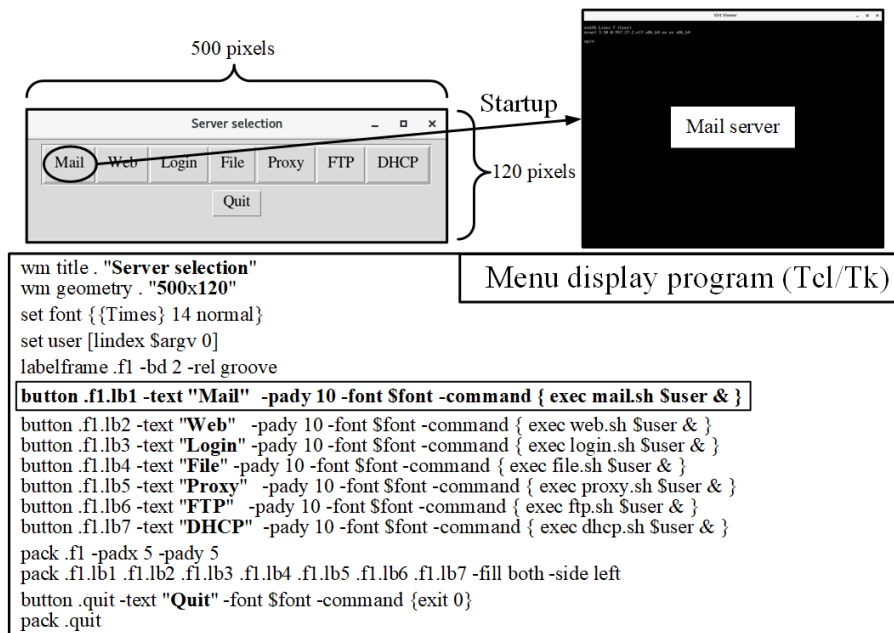


Fig. 3: Menu display of Tcl/Tk composed program.

Because the SSCTS is designed to be used by multiple users at the same time, it was necessary to prepare system software and virtual servers that can be operated by individual users. Therefore, it has a system configuration that can manage them efficiently. Figure 4 shows the system software configuration and its features. Note that only the program for displaying the main menu is distributed to each user. The configuration required to use the program, which executes each menu function and is owned by the server administrator (root), is different for each user. To allow each user to operate the virtual server differently in, e.g., the server construction exercise, the username is examined at login, and the directory path for the virtual server system data is changed according to the username.

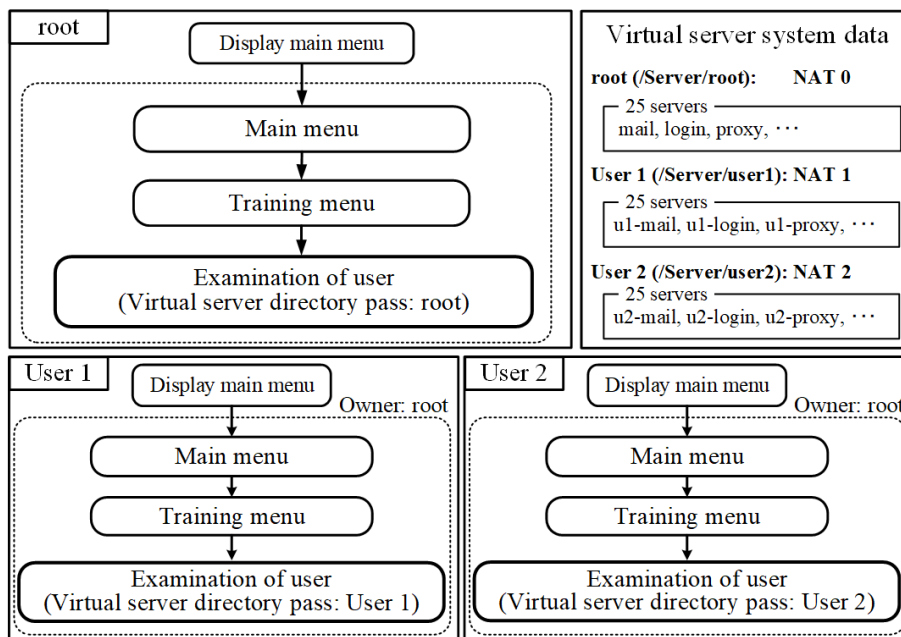


Fig. 4: System software configuration and features.

In the figure, we can see that when using the mail server, User 1 is set to use u1-mail belonging to the NAT 1 network, and User 2 is set to use u2-mail belonging to the NAT 2 network. In the SSCTS, 25 virtual servers are assigned to each user, who is expected to use them to execute the training menu shown in Fig. 2(b). Because the maximum disk capacity of one virtual server is 10 GB, the maximum disk capacity required for one user is 250 GB.

Since the SSCTS prepares 25 virtual servers for educating each user, in order to prepare an environment that can be used by multiple users to use at the same time, it is necessary to copy and configure 25 virtual servers each time a new user is created. Generally speaking, in a system with multiple virtual servers as one unit, it takes a long time to port the system or conduct other expansions. In addition, the potential for human error to introduce flaws due to manual operation is a matter of serious concern. To minimize these problems, we designed the automatic porting system shown in Fig. 5. To set a virtual server with the same functions for each user, a virtual local area network (LAN) is constructed using a different NAT for each user.

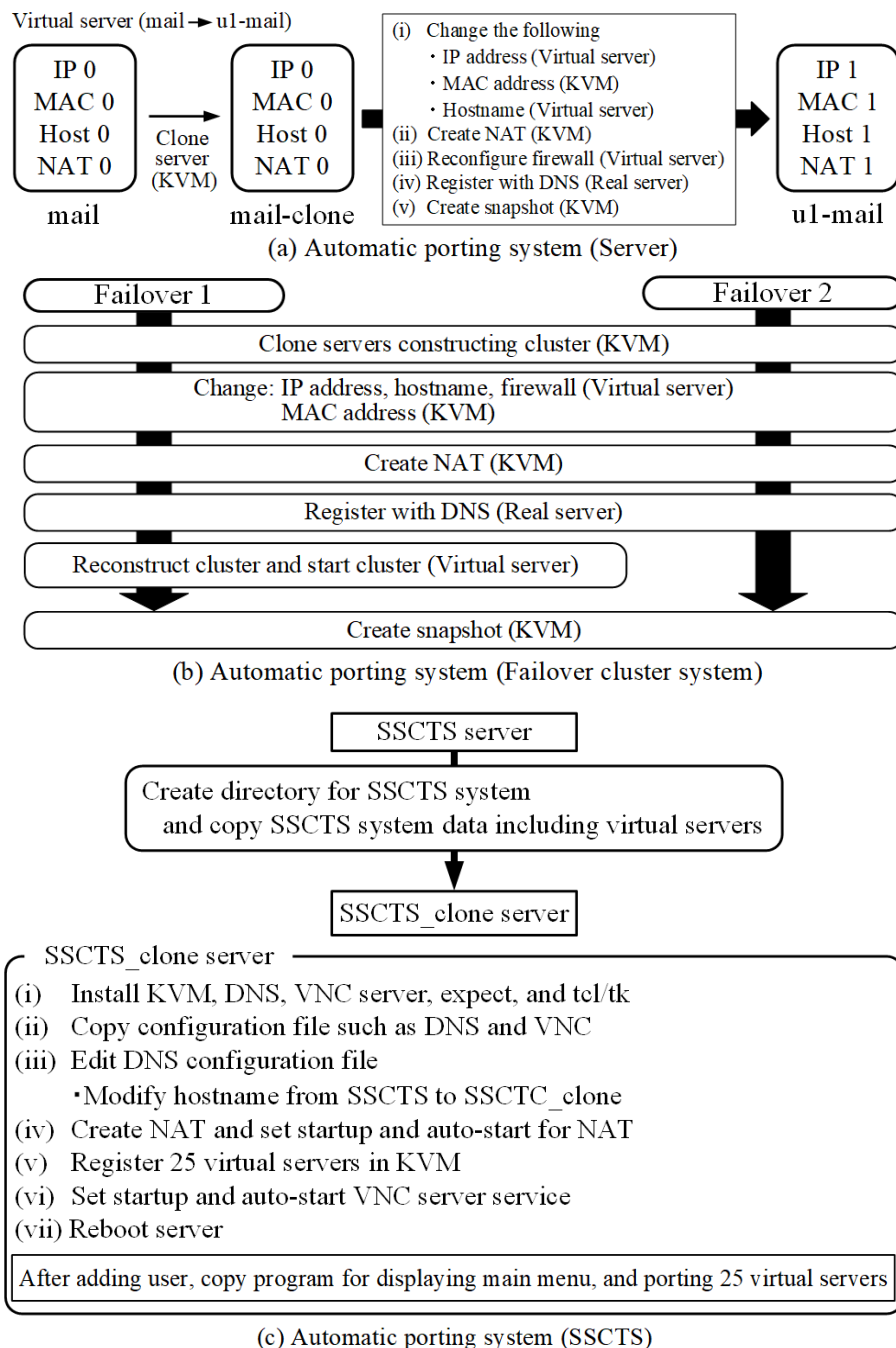


Fig. 5: Automatic porting system details.

The procedure for porting the virtual server is shown in Fig. 5(a). Here, the cloning source server is mail, and the ported server is u1-mail. The clone server mail-clone is created from the mail server and has the same settings as the mail server, such as the Internet Protocol (IP) address. Thus, u1-mail can be created by executing the settings shown in (i) through (v). As an example, Fig. 5(b) shows the procedure for porting the server system using the FOC system. Two servers that configure the FOC system are cloned, and the processing shown in Fig. 5(a) is performed. Next, the cluster is reconstructed in the two servers, and the cluster operation starts. Finally, a snapshot is created.

A porting experiment using the SSCTS server, which has a Core i9-9900K central processing unit (CPU), 64 GB of memory, and a solid-state drive (SSD) as storage media, was conducted. The CentOS standard has a file descriptor of 1024, a maximum number of 256,221 processes, and unlimited total CPU time and memory usage. The standard OS settings are used. Here, it should be noted that if this porting operation were to be performed manually, it would take approximately 0.5 hours for one virtual server. In contrast, one virtual server can be completed in approximately 1 minute with the proposed system, and there is no need to worry about human errors. Based on a calculation of 25 virtual servers, the creation process for each user takes approximately 12.5 hours and 0.4 hours for manual operation and the proposed system, respectively. This means that if the proposed system is adopted, the time required for porting operation is approximately 1/30 of the time required for manual porting.

Figure 5(c) shows the SSCTS automatic porting system. In this figure, the SSCTS server is the computer on which the system is installed, and the SSCTS_clone server (on which only an OS is installed) is the porting target. First, the SSCTS system data, including the 25 virtual servers, are copied from the SSCTS server to the SSCTS_clone server. Next, the SSCTS clone server can be constructed by executing the settings shown in (i) through (vii). A porting experiment using the SSCTS_clone server, which has a Core i7-3770 CPU, 8 GB of memory, and an SSD, is conducted. Here, standard CentOS settings are used, and it can be seen that the porting operation was completed in approximately 0.25 hours. However, because only the administrator (root) is set in this state, before it can actually be used, it is necessary to first add the users, then copy the program for displaying the main menu (shown in Fig. 4) before finally porting the 25 virtual servers.

4. Server system construction training

4.1. Server monitoring

From the Monitoring menu, exercises for two monitoring types, external monitoring using a client and internal monitoring logged in to the target server, can be targeted at the sample server. In addition, monitoring using Zabbix can be performed.

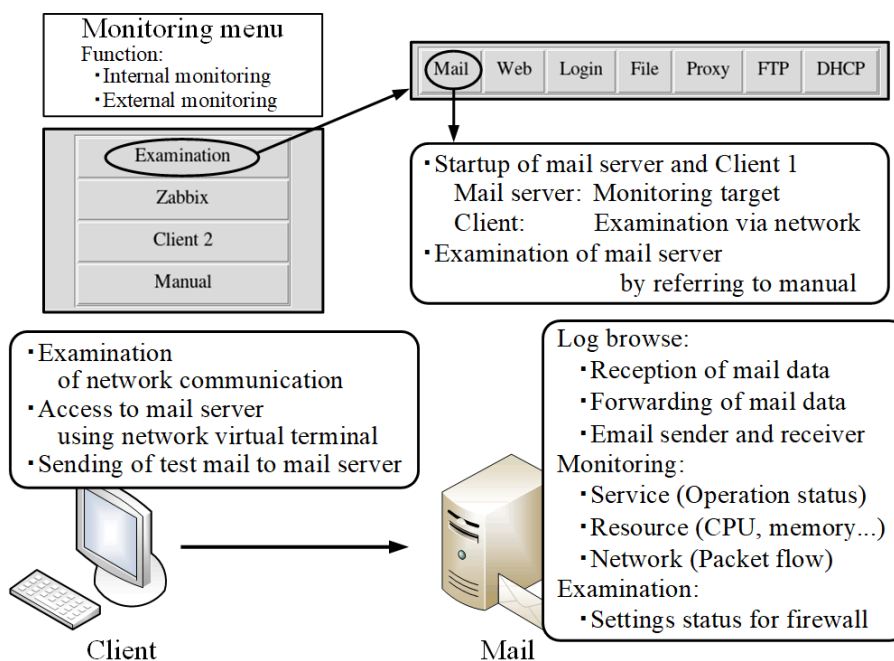


Fig. 6: Server monitoring exercises.

As stated previously, Client 2 handles the mail, file, proxy, and DHCP servers and is used for examining their settings and statuses. The system construction manual describes how to examine the log file of each server, how to use the commands necessary to monitor and examine them, how to examine the setting status in Client 2, and how to use Zabbix. Figure 6 shows an example of monitoring and examining the mail server. When selecting “Examination” in the Monitoring menu, the available sample server menu selections are displayed. Here, when selecting “Mail”, the mail server and Client 1 are started. External monitoring activities that can be performed from the client include network access by using the UNIX “ping” command to the mail server, examining the service offer status using the virtual network terminal, sending test mail to the mail server, and automatically examining the service offer status using the Expect language. Internal monitoring activities that can be performed include examining the log that records the receiving and forwarding mail statuses, monitoring the operating status of the program that provides services, monitoring the detailed status of resource usage and network access, and examining the firewall setting status. Similar examinations for servers other than the mail server are also available.

4.2. Server construction

In the Server Construction menu, construction can be performed while referring to the manual, which describes all the necessary settings, commands, and usage methods. Figure 7 shows the construction of the mail server as an example. If “Training” in “Mail” is selected from the Server Construction menu, training server startup is performed, and the manual for constructing the mail server is displayed. Here, it should be noted that installing the mail forwarding program (postfix) and the receiving program (dovecot) can also be performed while referring to the manual. Next, the processes of editing the configuration file required for each program, the firewall settings, and the startup and auto-start settings for each program are performed. To determine whether the constructed mail server can operate normally, the method explained in the Monitoring menu can be used.

For servers other than the mail server, httpd for the web server, sshd for the login server, nfs for the file server, squid for the proxy server, vsftpd for the FTP server, and dhcpd for the DHCP server were installed, and examinations of the settings and operating statuses of those servers are performed in the same way as the mail server. The method explained in the Monitoring menu can be used to determine whether the constructed server operates normally.

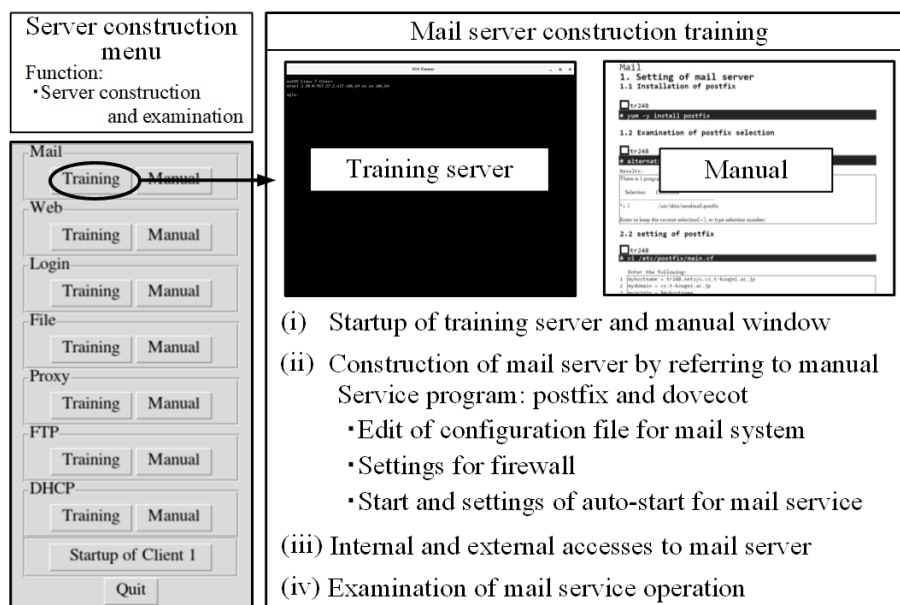


Fig. 7: Server construction.

4.3. Server system construction

In the System Construction menu, construction exercises (Training) and operating status examinations for the server system (Examinations) can be performed while referring to the manual. Here, it is possible to perform exercises related to seven systems that are often adopted in general server systems. Figure 8(a) shows the FOC system that allows another server to continue providing a service even if one server fails, while Fig. 8(b) shows a load balancing cluster (LBC) system that automatically disconnects a server when it fails and then provides services to the client via a server that is operating normally. Figure 8(c) shows a site-to-site VPN system that enables communications with a private IP address at remote locations via the Internet, while Fig. 8(d) shows a remote access VPN system that can connect a client installing the VPN client software to the base site LAN. In Fig. 8(e), we see a monitoring system that can monitor servers and network devices; Fig. 8(f) shows an Ansible system that can construct multiple servers with similar functions, and Fig. 8(g) shows a redundant load balancer system.

In the system construction exercises, Pacemaker and Corosync are installed to the FOC system, as shown in Fig. 8(a); HAProxy is installed to the LBC system shown in Fig. 8(b) and the redundant load balancer system shown in Fig. 8(g). In addition, SoftEther is installed in the VPN system, as shown in Figs. 8(c) and (d); Zabbix is installed to the monitoring system, as shown in Fig. 8(e); and Ansible is installed in the Ansible system, as shown in Fig. 8(f). Here, the configuration file edits required for each program, the firewall settings, and the auto-start startup and settings for each program are performed. Thereafter, the system operating status is examined by the method explained in the Monitoring menu.

Next, the system configuration used in the construction exercise is explained. In the FOC system, one client, one file server, and two installation targeted servers (Failover 1 and Failover 2) are used, as shown in Fig. 8(a), while one client, one file server, one load balancer server, and two redundancy servers are used in the LBC system shown in Fig. 8(b). Additionally, one client, one server, two VPN servers, and one router are used in the site-to-site VPN system shown in Fig. 8(c), while one client, one server, one VPN server, and one router are used in the remote access VPN system shown in Fig. 8(d). Furthermore, one client, one Zabbix server, and six sample servers for use as target servers (excluding the file server) are used in the Zabbix-based monitoring system shown in Fig. 8(e), while one client, one Ansible server, and two installation targeted servers are used in the Ansible system shown in Fig. 8(f). Finally, one client, one file server, two load balancer servers, and two redundancy servers are used in the redundant load balancer system shown in Fig. 8(g). Here, the router is constructed using a training server.

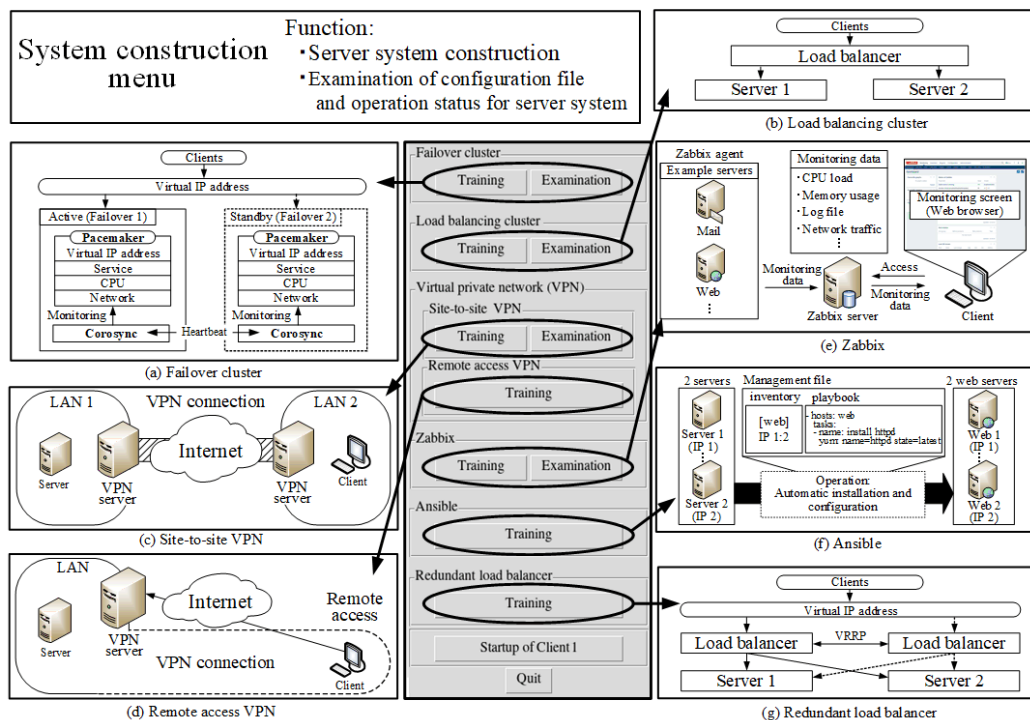


Fig. 8: Server system construction.

4.4. Troubleshooting

Figure 9 shows the function and mechanism of the Troubleshooting menu. Here, users can select either “Manual selection”, which is used to manually select the problem server and the fault type, or “Automatic selection”, which uses random numbers to automatically select the problem server that causes the fault and the fault type. When automatic selection is used, fault isolation processing exercises can be conducted. When manual selection is used, it is first necessary to select the faulty server and then select the fault type. Troubleshooting exercises are available for stopping the service-providing program, stopping the program that performs network communication, and security system faults such as the firewall program stoppage and setting the permissions to prevent unauthorized network access. After executing the fault, the abnormal state is examined, and troubleshooting exercises are conducted while referring to the manual.

In the automatic selection process, the number of targeted servers for troubleshooting is set first. Here, “One”, “Two”, or “All” can be selected. The targets are a mail server in the case of “One”, a mail server or web server in the case of “Two”, and five servers in the case of “All”. In the cases of “Two” or “All”, one of the faults is selected by a random number. Here, it should be noted that the random numbers used in this system, which are also used to determine the fault type, are determined by the seed value. More specifically, to complexly combine defective servers and fault types, the execution time is used for the seed value, and the random numbers themselves are generated as shown in the figure. Taking the random number R as an example, when execution is performed at 10:15:30, 11 random numbers are generated with the seed value set to 31, and the 11th random number value is used as R . The number of the faulty server shown in the figure is the value obtained by dividing the random number R by five and then adding one to the remainder.

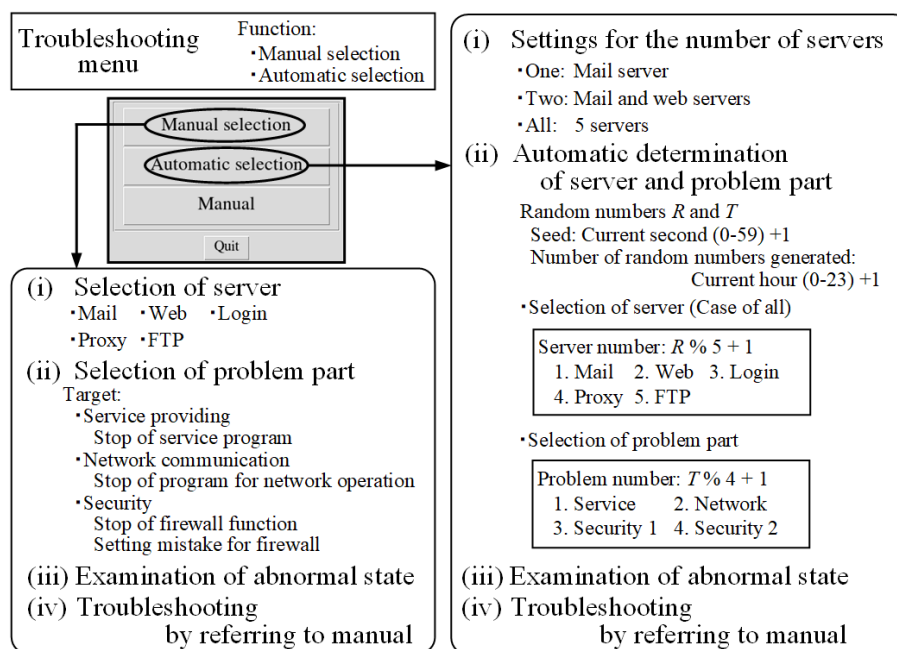


Fig. 9: Troubleshooting.

5. Experiments to examine comprehension levels

To examine the user comprehension levels following perusal of the education contents presented by the SSCTS, a comparative experiment using the SSCTS and a real server system was conducted, taking the construction of an FOC system as an example. To prepare for the experiment, it was first necessary to create a system operating manual, an FOC system construction manual, test questions for examining the user comprehension levels, and a questionnaire. It was also necessary to construct a real server system that could be used for initializing settings.

5.1. Experiment preparations

In this experiment, the users were required to construct the FOC while using the server system in the initial state. Therefore, an initialization system was required. Figure 10 shows the flow of the FOC construction and the initialization procedure of each system. The FOC construction flow is shown in Fig. 10(a). PCS, which is a Corosync and Pacemaker configuration tool, and Apache were installed in two servers (Failover 1 and Failover 2), and PCS was set to the operating state. Next, the firewall was set in order to enable communications between the two servers. Because those two servers offer the same service to the client, they connect to the file server through a Network File System (NFS) connection as a shared area. FOC construction was then completed by constructing the cluster and setting the resources. The procedure for initializing the FOC system in the SSCTS is shown in Fig. 10(b). The two virtual servers are then shut down, and complete initialization is realized by reverting those servers to the state creating the snapshot. FOC system initialization in the real server system is shown in Fig. 10(c), where it can be seen that initialization is performed by setting (i) through (v). However, the server cannot be initialized if an operation other than the settings shown in Fig. 10(a) is performed.

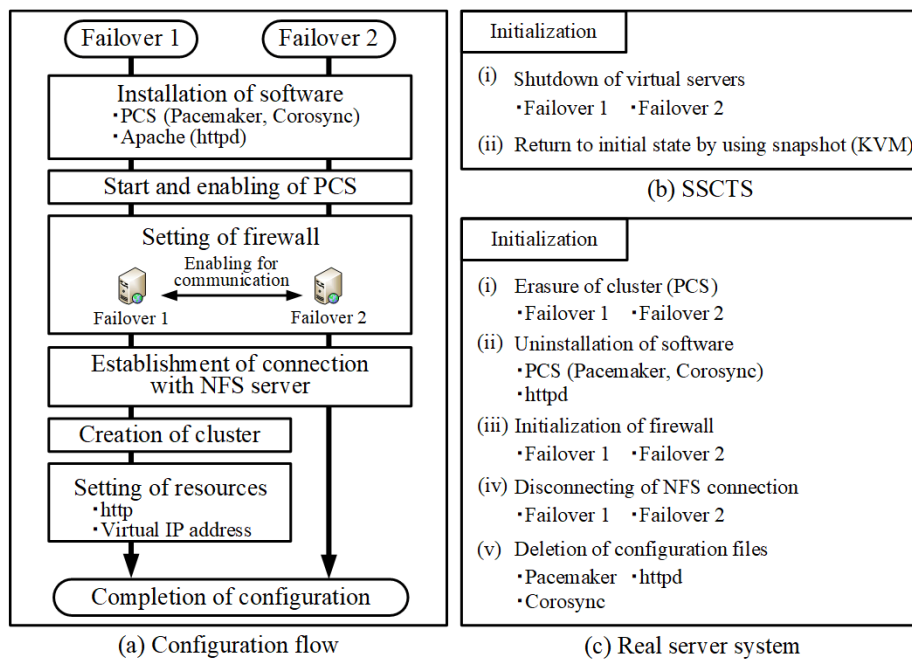


Fig. 10: Flow of FOC construction and initialization of each system.

The system operating manual describes SSCTS menu operations, how to start the real server system, and how to log in to each system. Figure 11 shows the creation of the FOC system construction manual. Both the manuals and the test questions were created based on the educational objectives considered to be important for constructing the FOC. The training objectives for the outline and FOC system construction are shown in Fig. 11(a). When creating this outline, it was necessary to ensure readers would understand the necessity of redundancy and the roles of the software, as well as the need for the virtual IP address that would be used to construct the FOC. In addition, during construction, the installation of FOC software, its setting method, and the firewall settings are critically important.

The manual structure is shown in Fig. 11(b). A video was adopted for the FOC outline, and a paper medium was adopted for the construction manual. Construction manual explanations and execution commands were described based on the training objectives. In addition, as shown in Fig. 10(a), since there are cases where construction is performed using two servers and cases where it is performed with just one server, a check box is provided to prevent the user from forgetting to make the appropriate setting. Furthermore, in order to make input commands easier to see, black and white were inverted, and the font size was set to 14 points. The Google Forms software suite was adopted for the test, which consisted of 15 multiple-choice questions (five choices per question). The number of questions set for each test content is as follows:

- FOC outline (four questions)
- Software to configure (two questions)
- Commands used during construction (five questions)
- Settings related to construction (two questions)
- Operating status examinations (two questions)

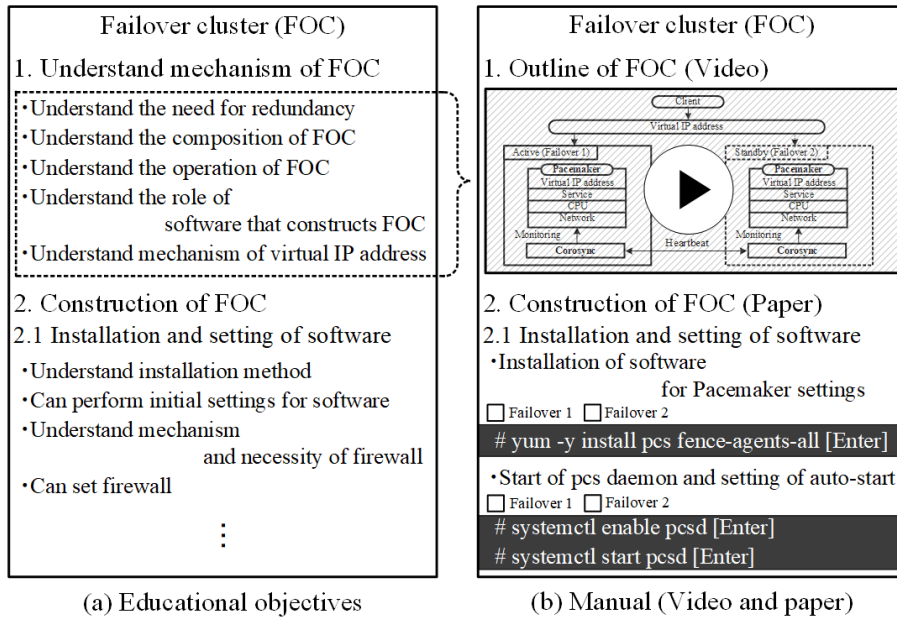


Fig. 11: FOC system construction manual creation.

5.2. Experimental system configuration, specifications, and experimental procedure

Figure 12 shows the configuration of the experimental system and the experimental procedure.

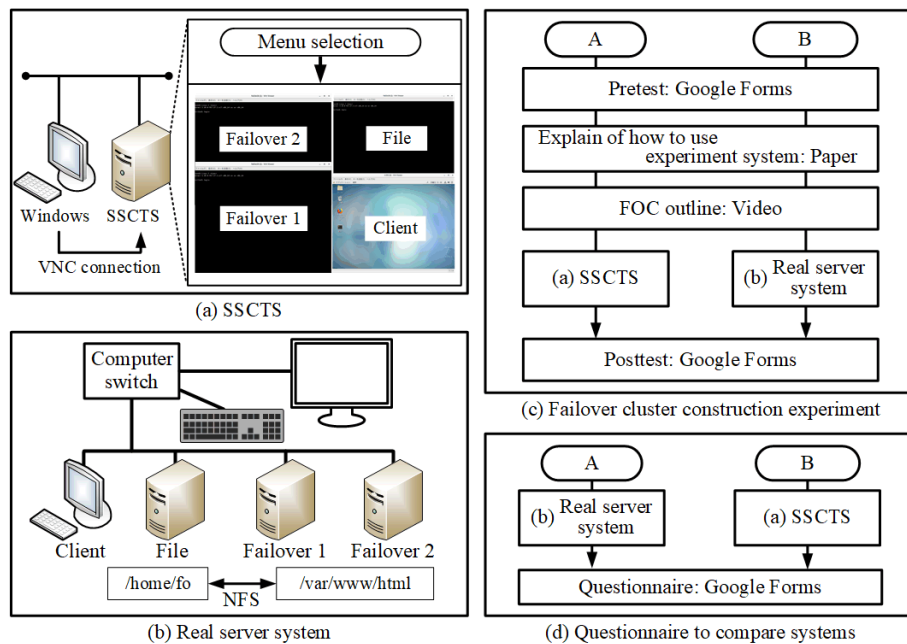


Fig. 12: Experimental system configuration and experimental procedure.

The SSCTS configuration is shown in Fig. 12(a). The SSCTS experimental system consists of a computer with the Windows OS installed and a computer with the SSCTS installed. Figure 12(b) shows the configuration of the real server system, which consisted of a client to execute access examinations, a file server for data sharing, two servers for FOC construction, and one computer switch. The experimental procedure is shown in Fig. 12(c). The experiment was conducted by SSCTS users (Group A) and real server system users (Group B). The users were 3rd- and 4th-year university students who had completed the UNIX exercises of class subjects in our university and had no experience in server construction. Here, it should be noted that to ensure the conditions for both groups were similar, the test contents, manuals, and questionnaires were the same for both groups.

Before performing system construction, pretests were performed on both groups' users to examine their current level of understanding regarding the FOC. Thereafter, the explanation manual for operating each system and the manuals necessary for constructing the FOC were distributed and described. Members of both groups then watched a video explaining the FOC outline, after which they actually constructed the FOC. Upon completion, both groups were given a posttest to examine their improved comprehension of the FOC. After both groups had completed the test, the systems were switched, and both groups again constructed the FOC, as shown in Fig. 12(d). Finally, all users in both groups were asked to complete the operability questionnaire.

Table 1 shows the experimental system specifications. The SSCTS shown in the upper portion of the table is equipped with a Core i9 9900K CPU and 64 GB of memory to ensure it can simultaneously connect to multiple users. Standard CentOS settings are used. KVM is used as the server virtualization technology, a graphical user interface (GUI) is used for the virtual client, and a character-based user interface (CUI) is used for the virtual server in order to utilize memory effectively. The real server system (shown in the lower half of the table) uses CUI for the servers that construct the FOC and the file server. When the FOC system is operating, the SSCTS power consumption is 50.1 W, while that of the real server system is 168.0 W.

Table 1: Experimental system specifications.

SSCTS					
Specification	Real server		Specification	Virtual client and servers	
				Client	File
CPU	Core i9 9900K (3.60 GHz / TB: 5.00 GHz)		CPU	1 Core	
Memory	64 GB		Memory	1 GB	
System software	DNS, VNC server, Tcl/Tk, expect		Console	GUI	CUI
Virtualization software	KVM 1.5.3		System software	nfs	Pacemaker, Corosync, Apache
OS	CentOS 7.9		OS	CentOS 7.9	
Power	50.1 W				

Real server system			
Specification	Client	File	Failover 1, 2
CPU	Core 2 Duo E4500 (2.20 GHz)	Core i7 3770 (3.40 GHz / TB: 3.90 GHz)	Core i5 10400 (2.90 GHz / TB: 4.30 GHz)
Memory	4 GB	8 GB	
System software		nfs	Pacemaker, Corosync, Apache
Console	GUI	CUI	
OS	CentOS 7.9		
Power	168.0 W		

5.3. Experimental results and their analysis

Table 2 shows the experimental results of the comprehension examination for Groups A and B, and their analysis of variance (ANOVA) results. In this table, "Difference" refers to the difference between the pretest and posttest results. A total of 16 users, eight users in each group, participated in the experiment, which consisted of two tests, two system constructions, and one questionnaire. The experimental procedure took each user approximately 2 hours. ANOVA was performed based on the pretest and the posttest results for each user. The Factor T (test) showed a significant difference between the pretest and posttest results ($p < 0.001$), thus indicating improvements in the scores of both groups. Additionally, from the average value of the score difference, it can be seen that the level of improvement was the same for both groups, and there was no significant difference in the factor G (group) because the value F was close to 1.

Table 2: Experimental results of comprehension examination.

SSCTS (Group A)				Real server system (Group B)			
User	Pretest	Posttest	Difference	User	Pretest	Posttest	Difference
A1	47	73	26	B1	33	67	34
A2	40	67	27	B2	47	60	13
A3	53	60	7	B3	33	60	27
A4	47	53	6	B4	33	40	7
A5	20	33	13	B5	47	73	26
A6	33	73	40	B6	33	40	7
A7	47	53	6	B7	33	53	20
A8	33	87	54	B8	27	67	40
Average	40.0	62.4	22.4	Average	35.8	57.5	21.8
Standard deviation	10.1	15.4	16.6	Standard deviation	6.78	11.5	11.4

Factor G (Between-subjects design): Group Number of levels: 2 (Groups A, B)	Source	<i>F</i> value	<i>p</i> value
	Factor G	0.925	0.353
	Factor T (Within-subjects design): Test Number of levels: 2 (Pretest, posttest)	33.6	0.000
		Factors GT	0.007 0.936

Table 3 shows the pretest and posttest scores. Each score was standardized for each test content for a maximum score of 100. Moreover, their differences and the ANOVA results are shown.

Table 3: Score analysis by test content.

SSCTS (Group A)			
Analysis items (Number of questions)	Pretest	Posttest	Difference
1. Outline for FOC (4)	65.6	65.6	0.00
2. Software to configure (2)	18.8	68.8	50.0
3. Commands used to construct (5)	40.0	62.5	22.5
4. Settings for constructing (2)	25.0	62.5	37.5
5. Examination of operating conditions (2)	25.0	50.0	25.0

Real server system (Group B)			
Analysis items (Number of questions)	Pretest	Posttest	Difference
1. Outline for FOC (4)	59.4	53.1	-6.3
2. Software to configure (2)	18.8	50.0	31.2
3. Commands used to construct (5)	42.5	65.0	22.5
4. Settings for constructing (2)	18.8	62.5	43.7
5. Examination of operating conditions (2)	6.3	56.3	50.0

Factor G (Between-subjects design): Group	Number of levels: 2 (Groups A, B)		
Factor T (Within-subjects design): Test	Number of levels: 2 (Pretest, posttest)		
Factor I (Within-subjects design): Analysis item	Number of levels: 5 (Analysis items 1 - 5)		

Source	<i>F</i> value	<i>p</i> value
Factor G	1.381	0.260
Factor T	39.612	0.000
Factors GT	0.020	0.888
Factor I	3.391	0.015
Factors GI	0.182	0.947
Factors TI	4.413	0.004
Factors GTI	0.821	0.518

Simple main effect in interaction (Factors TI)		
Effect	<i>F</i> value	<i>p</i> value
T(I1)	0.118	0.733
T(I2)	19.873	0.000
T(I3)	6.096	0.016
T(I4)	19.873	0.000
T(I5)	16.933	0.000

Because the factor T (test) shows a significant difference ($p < 0.001$), it is clear that the posttest results indicate an overall improvement over the pretest results. Additionally, because a significant difference ($p < 0.05$) is seen in factor I (analysis item), it is clear that score differences exist for each item. Furthermore, since factor TI (the interaction between the test and the analysis item) shows a significant difference ($p < 0.005$), we can conclude that the posttest results, as a whole, improved for each item. However, the results also show that there were no significant differences in analysis item 1 in the simple main effect in the interaction of factors TI. In this case, analysis item 1 is the FOC outline, while the other items involve actual construction and operation issues. Taken together, these results show improvements in the user knowledge for the learning portions that relate to actual FOC construction and operation.

The questionnaire results did not identify any operability issues, and similar comprehension improvements can be observed in the SSCTS and real server system environments. Furthermore, the SSCTS is superior in other cost-related aspects because the number of servers required for constructing the system is small, and since a virtual server is used, it can be initialized reliably.

6. Conclusions

This paper reported on the development of a server system construction training system (SSCTS) that can be used simultaneously by multiple users on a single computer because it adopts server virtualization technology and virtual network computing. Because the basic operations of the SSCTS are handled via a menu selection method, the UNIX “Tcl/Tk” toolkit was used for its development. Additionally, because 25 virtual servers are set in the SSCTS for each user, we developed and introduced an automatic porting system to shorten the time required to add users and prevent human errors. The “active learning” method was adopted, and it was possible to conduct exercises related to monitoring, server construction, system construction, and troubleshooting, all of which are necessary for server system construction, operation, and management.

A comprehension examination experiment using the construction of a failover cluster system as an example was conducted. We developed a real server system that could be initialized for comparison with the SSCTS. The experiment was conducted with a total of 16 users split into two groups of eight users: Group A using the SSCTS and Group B using the real server system. Tests were conducted before and after the system construction exercises by both groups in order to determine comprehension level improvements, and a similar level of improvement was observed in the results of both groups.

It should also be noted that other cost-related aspects were minimized because the number of servers required for constructing the system is small. Additionally, since a virtual server is used, it can be initialized reliably. This indicates that the SSCTS has a system advantage over real server systems in terms of the number of servers and the system initialization required for a learning system.

However, although it is possible to learn basic server system construction techniques with SSCTS, it is not possible to address security issues such as cracking and denial-of-service (DoS) attacks with the current version because they require countermeasures that vary depending on the situation. As a future study, we will analyze the learning histories obtained and consider learning effect improvement methods based on those results.

References

- [1] Y. Liu, A. Liu, N. N. Xiong, T. Wang, and W. Gui, “Content Propagation for Content-Centric Networking Systems From Location-Based Social Networks,” *IEEE Trans. on Systems, Man, and Cyber.: Systems*, Vol. 49, No. 10, 2019, pp. 1946-1960.
- [2] Information-technology Promotion Agency (JAPAN), <https://www.ipa.go.jp/files/000085256.pdf>, 2020.2.10.
- [3] J. Xiao, B. Wu, L. Zhang, H. Wen, X. Jiang, and P.-H. Ho, “Joint Design on DCN Placement and Survivable Cloud Service Provision over All-Optical Mesh Networks,” *IEEE Trans. Commun.*, Vol. 62, No. 1, 2014, pp. 235-245.
- [4] Q. Zhang, M.-F. Zhani, M. Jabri, and R. Boutaba, “Venice: Reliable Virtual Data Center Embedding in Clouds,” *Proc. IEEE INFOCOM 2014*, Vol. 1, 2014, pp. 289-297.
- [5] M.-G. Rabbani, M.-F. Zhani, and R. Boutaba, “On Achieving High Survivability in Virtualized Data Centers,” *IEICE Trans.*

- Commun., Vol. E97-B, No. 1, 2014, pp. 10-18.
- [6] T. Ono and K. Ueda, "Data Redundancy Dynamic Control Method for High Availability Distributed Clusters," ACM Proceedings, No. SoICT 2018, 2018, pp. 185-191.
- [7] M. Kitamura, "Configuring a Low-cost, Power-saving Multiple Server Backup System: Experimental Results," IEICE Trans. Commun., Vol. E95-B, No. 1, 2012, pp. 189-197.
- [8] M. Kitamura, Y. Udagawa, H. Nakagome, and Y. Shimizu, "Development of a Server Management System Incorporating a Peer-to-Peer Method for Constructing a High-availability Server System," JISSJ, Vol. 13, No. 2, 2018, pp. 14-40.
- [9] M. Kitamura, Y. Shimizu, and K. Tani, "Development and Operation Experiment of a Power-saving, High-availability Server System by Compound Operation of a Power-saving Server System and a Multiple-server Backup System," JISSJ, Vol. 15, No. 2, 2020, pp. 34-54.
- [10] A. A. Ben, M. Abid, and A. Meddeb, "Secure Fog-Based E-Learning Scheme," IEEE Access, Vol. 8, 2020, pp. 31920-31933.
- [11] S. Jiang, K. Yang, and Y. Chen, "Analysis on the Construction of Personalized Teaching System Based on Cloud Computing Platform," IEEE Conference Proc., Vol. 2020, No. CVIDL, 2020, pp. 518-522.
- [12] S. Wang, L. Zhu, and M. Cheng, "Docker-based Web Server Instructional System," IEEE Conference Proc., Vol. 2019, No. ICIS, 2019, pp. 285-289.
- [13] W. Xiao, "Quality-Centered Mobile Terminal Teaching Assessment System under the Background of Big Data," IEEE Conference Proc., Vol. 2020, No. ICSGEA, 2020, pp. 446-449.
- [14] D. B. Domingueti, D. R. C. Dias, and D. B. F. Carvalho, "AVRA: virtual learning environment with virtual reality capabilities," IEEE Conference Proc., Vol. 2020, No. SVR, 2020, pp. 102-110.
- [15] T. Otsuka and H. Tamaki, "ProBE: a learning support system for programming," IEICE Technical Report, Vol. 119, No. 236, 2019, pp. 45-50.
- [16] N. Tanyakomdilok, A. Miyatake and M. Sasayama, "e-Learning System for Programming Exercise," IEICE Technical Report, Vol. 119, No. 468, 2020, pp. 127-130.
- [17] Y. Kobayashi and S. Hasegawa, "A Proposal for Learning Environment to Evaluate Computational Thinking for Elementary School Students in Japan from Operation Point of View," IEICE Technical Report, Vol. 120, No. 167, 2020, pp. 13-18.
- [18] T. Iwasaki, Y. Tateiwa, and T. Yasuda, "A training environment for server OS installation on a support system for network administrator education based on virtual machine technology," IEICE Technical Report, Vol. 109, No. 82, 2009, pp. 27-32.
- [19] Y. Ooka, Y. Tateiwa, and N. Takahashi, "Development of the network collaboration construction exercise system using virtual machine," IEICE Technical Report, Vol. 116, No. 484, 2017, pp. 223-228.
- [20] N. Hoshikawa, Y. Kobayashi, M. Ishihara, A. Shiraki, T. Shimobaba, and T. Ito, "Implementation of Denial of Service Attack Exercise System and Education of Security Technology by the Active Learning Scenario," IEICE Trans., Vol. J103-B, No. 4, 2020, pp.180-183.
- [21] M. Yukawa, Y. Taniguchi, and N. Iguchi, "A System for Supporting Learning of Network Security Through Offensive and Defensive Battle Exercise," IEICE Trans., Vol. J103-D, No. 8, 2020, pp.591-602.
- [22] D. Suzuki, "Practice and Evaluation of a Penetration Testing Exercise for Raising Cybersecurity Awareness," IEICE Technical Report, Vol. 118, No. 152, 2018, pp.131-136.
- [23] K. Nisimura and H. Igaki, "Consideration of User Behavior History Collection System for Server Management Exercise," IEICE Technical Report, Vol. 117, No. 380, 2018, pp. 155-160.
- [24] D. Suzuki, "Server Configuration Training Including Vulnerability Assessment and Remediation process," IEICE Technical Report, Vol. 119, No. 140, 2019, pp. 259-263.
- [25] Y. Hoshino, K. Notomi, H. Nishimura, and H. Shimeno, "A Development and Evaluation of Training Environment for System Administrator Based on Linux Server," IEEJ Trans. on Electronics, Information and Systems, Vol. 136, No. 7, 2016, pp. 986-994.

[26] Information-technology Promotion Agency (JAPAN), <https://www.ipa.go.jp/files/000025010.pdf>, 2021.9.8.

[27] Information Technology Curricula 2017,
<https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf>, 2021.9.8.

Authors Biography

Mitsuyoshi KITAMURA

He received his B.E. degree from Tokyo Polytechnic University in 1984. In 1990, he became a research assistant at Tokyo Polytechnic University, where he became an assistant professor in 2003. His current interests are the optimum design and construction of server-client systems and the analysis of various characteristics of server systems.

Takumi YAMAGUCHI

He received his B.E. degree in 2020 from Tokyo Polytechnic University, where he is currently working toward the M.S. degree. He is engaged in developing software for experimental systems and analyzing performance data.

Toshikazu TAKESHITA

He received his B.E. degree in 2020 from Tokyo Polytechnic University, where he is currently working toward the M.S. degree. He is in charge of implementation of experimental hardware systems and analyses of power-saving data.

Takahito TOMOTO

He received his PhD (Engineering) from Hiroshima University. His current occupation is Associate Professor at Tokyo Polytechnic University. He is engaged in research on educational technology and intelligent learning support systems.