computers

MDPI

Article

Experiments and Evaluation of a Container Migration Data-Auditing System on Edge Computing Environment

Toshihiro Uchibayashi [1,*], Bernady Apduhan [2], Takuo Suganuma [3] and Masahiro Hiji [4]

[1] Research Institute for Information Technology, Kyushu University, Fukuoka 816-8580, Japan
[2] Faculty of Information Science, Kyushu Sangyo University, Fukuoka 813-0004, Japan
[3] Cyberscience Center, Tohoku University, Sendai 980-8577, Japan
[4] Faculty of Economics, Tohoku University, Sendai 980-8577, Japan
* Correspondence: uchibayashi.toshihiro.143@m.kyushu-u.ac.jp

Abstract: With the proliferation of IoT sensors and devices, storing collected data in the cloud has become common. A wide variety of data with different purposes and forms are not directly stored in the cloud but are sent to the cloud via edge servers. At the edge server, applications are running in containers and virtual machines to collect data. However, the current deployment and movement mechanisms for containers and virtual machines do not consider any conventions or regulations for the applications and the data it contains. Therefore, it is easy to deploy and migrate containers and virtual machines. However, the problem arises when it is deployed or migrated, which may violate the licensing terms of the contained applications, the rules of the organization, or the laws and regulations of the concerned country. We have already proposed a data-audit control mechanism for the migration of virtual machines. The proposed mechanism successfully controls the unintentional and malicious migration of virtual machines. We expect similar problems with containers to occur as the number of edge servers increases. Therefore, we propose a policy-based data-audit control system for container migration. The proposed system was verified in the implemented edge computing environment and the results showed that adding the proposed data-audit control mechanism had a minimal impact on migration time and that the system was practical enough. In the future, we intend to conduct verification not in a very compact and short-range environment such as this one but on an existing wide-area network.

Keywords: container; edge computing; migration; data-audit; policy

check for updates

Citation: Uchibayashi, T.; Apduhan, B.; Suganuma, T.; Hiji, M. Experiments and Evaluation of a Container Migration Data-Auditing System on Edge Computing Environment. *Computers* **2023**, *12*, 27. https://doi.org/10.3390/computers12020027

Academic Editor: Paolo Bellavista

Received: 30 November 2022
Revised: 16 January 2023
Accepted: 20 January 2023
Published: 27 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, hardware performance and power savings have improved significantly. It is now possible to execute complex calculations in small-size computers without the need for large computers and large amounts of power as in the past. In addition, sensors and devices that work in conjunction with these small-size computers have become remarkably popular. A wide variety of sensors and devices with different purposes and forms are available in the market, including sensors for measuring environmental conditions and wearable devices for measuring personal health conditions. As the miniaturization and performance of the computer has progressed, it has become common for multiple applications to be processed together in a single small-size computer. Furthermore, wrapping these applications in a virtual environment increases their versatility.

On an Internet of Things (IoT) application, data from sensors and devices are collected and stored data in the cloud via edge servers [1–3]. These edge servers handle a large volume of data including data containing personal information. The pre-processing at the edge server is performed before storing the data in the cloud, which reduces the volume of data and converts the data into data that will not be associated with individuals. Pre-processing is indispensable for handling large amounts of data at high speed. Pre-processing is performed at the edge server, as an IoT application service, that covers a wide

area and located near the IoT sensors/devices. Conducting experiments with thousands of sensors and/or devices can only be realized using virtual machines or containers in a virtual environment. Furthermore, migration of virtual machines or containers is essential for the efficient operation of virtual environments in edge computing environments (Figure 1).

Virtual machine and container migration has become an indispensable technology. However, the problem is that migration is simplified and can be easily performed by anyone who has the permissions. We have already proposed a data-audit mechanism for migrating virtual machines into the cloud. Virtual machine migration involves physically migrating a virtual machine running to another host machine without stopping it and is already supported by significant hypervisors such as KVM [4] and XEN [5]. Virtual machine migration execution is based only on the permissions and whether the migration is physically possible and does not consider the application and data regulations contained in the virtual machines. The problem arises when unintentional migration of virtual machines to the wrong destinations or the malicious migration of virtual machines. This can lead to violations of company rules and national laws. This is a severe problem. So, a control mechanism is needed to determine whether a virtual machine can be migrated, considering not only the permissions and physical possibility, but also the data that the virtual machine contains. This problem is the same for containers, meaning immediate countermeasures are needed.

Therefore, we will solve this problem by extending the control mechanism for virtual machine migration and apply it to containers. The container has virtual environments for each application. The container technology has improved dramatically over the past few years and is used not only in the field but is also the subject of much research and has attracted much attention. Here, we proposed a control mechanism of container migration and add the edge server attributes in data-auditing policy. Furthermore, we implement and demonstrate the usefulness of the proposed data-auditing mechanism as well as its verification on a virtual environment testbed in which the proposed mechanism was implemented.

The novelty of this proposal is that it focuses on the applications and data contained in containers, which have yet to be focused on, to prevent accidental or malicious migrations. Without human intervention, this proposal will ensure a certain level of security in orchestration environments, such as container system automation. This will further contribute to ensuring the security of container-based edge environments.

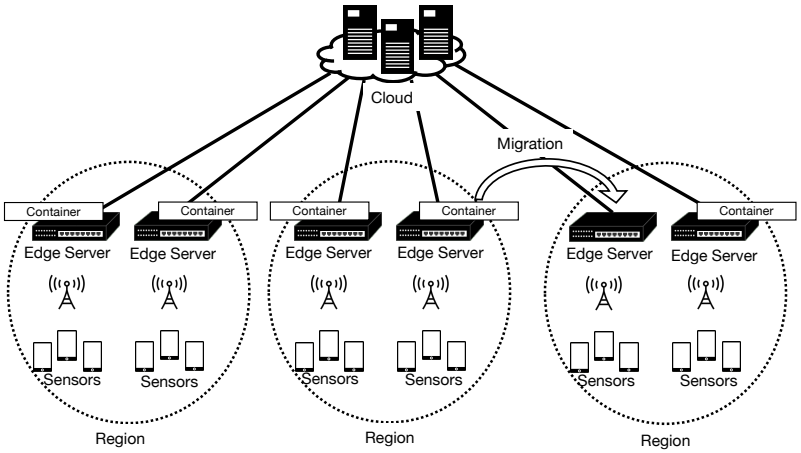

Figure 1. Container migration experiment in edge computing environment.

2. Related Works

2.1. Container Technology

Containers running through a container engine share the kernel of the host OS, isolating resources such as CPU and memory and creating a virtual space. In hypervisor-type virtualization, such as virtual machines, an OS must be installed separately in the virtual machine from the OS installed on the physical server, so when the virtual machine is started,

the OS must be started in the same manner as the physical server. In contrast, containers can run applications by sharing the kernel of the host OS, eliminating the need to start up a conventional OS when starting up a container (Figure 2). In other words, containers that contain only applications, middleware, and libraries are much lighter than virtual machines that contain the entire OS. Containers consume less processors and memory and use less storage, so they start up faster than virtual machines, and many containers can be run simultaneously on the same machine. In addition, container management software absorbs differences among hardware and operating systems, so containers that have already been tested are guaranteed to work even if the hardware is changed. For this reason, we conduct this research with containers. Furthermore, in edge computing environment, in which data is processed at nearby edge servers is becoming more prevalent, thus the demand for lightweight containers is increasing.

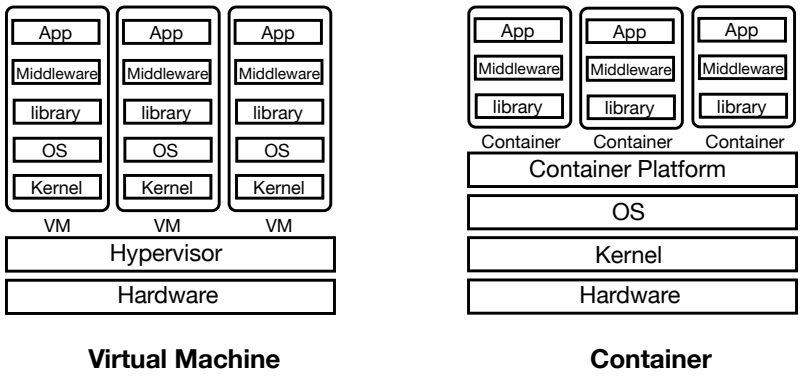

Figure 2. Structure of virtual machine and container.

2.2. Migration of Virtual Machine

This section introduces virtual machine migration and its policy control on cloud platforms. Virtual machine migration refers to the physical transfer of a virtual machine to another host machine. There are two types of migration: "live migration" and "cold migration (offline migration)". The virtual machine can be migrated in live migration while the process runs without stopping. This allows the OS running on the virtual machine, the application, and the data contained in the virtual machine to be migrated while still running. The memory image of the virtual machine running on the host machine is completely migrated to a virtual machine on another host machine. At the destination host machine, operations continue without stopping or disconnecting running applications. Although there is strictly a millisecond pause at the time of the switchover, the network session is never disconnected and the users of the applications running on the virtual machines are unaware that the switchover has occurred.

Research is being conducted to control this migration by applying various conditions and policies [6–11]. Role-based control is a method used to establish secure migration. Assuming a cloud environment wherein the current host machine, destination host machine, hypervisor, hardware, and communication channel are secure, add user controls to enable secure migration. In this approach, users who are authorized to perform migrations are defined by each virtual machine's policy. When a user instructs a virtual machine to perform a migration, the policy is used to check whether the user is authorized. This prevents information leakage due to migration by unauthorized users. However, this method does not prevent the risk of authorized users performing the migration in violation of data use consent conditions or privacy laws and regulations.

Whereas, the hardware requirement-based migration is a method used to perform virtual machine migration based on hardware requirements [12–17]. The destination host machine is determined according to the policy definition conditions of the hardware element. The host machine provider configures the virtual machines to use physical hardware

resources efficiently. However, the virtual machine must be migrated to another host machine if an overload is detected.

Likewise, in network configuration-based migration, the migration destination is determined in network configuration-based migration based on bandwidth, backbone network paths, and paths between related virtual machines [18–21]. Many applications provide services over the network. The backbone network of the host machine is a critical element for the virtual machine. Therefore, during migration, the host machine to be migrated is determined by the state of the backbone network.

There have been many studies on virtual machine migration control. However, since container technology has only been in the spotlight in recent years, research on container migration control still needs to be conducted.

2.3. Migration of Container

Containers are often used at the edge server, which is much less potent than in the cloud. Therefore, it is necessary to deploy containers to the most suitable edge server according to its purpose. Container migration is therefore attracting attention [22–24]. Container migration allows containers to be migrated to another edge server by freezing the current processing and then restarting at the destination server, thereby preserving the environment. Container migration at the edge server refers to the physical migration of containers to another edge server, just like the migration of virtual machines. As with virtual machines, there are two types of migration: "live migration" and "cold migration (offline migration)".

Typical container platforms offering migration, as of March 2022, are Docker [25], LXC [26], OpenVZ [27], and OpenShift [28]. Linux Container (LXC) is a system container that replaces KVM and Xen.

2.4. Data Protection for Container

This section presents research on data protection for containers.

Sultan et al. [29] provided a survey of literatures on container security and solutions. They also derived four general use cases that cover security requirements in the host and container threat landscape. The use cases are protection for applications in containers, protection between containers, host protection from containers, and container protection from malicious or semi-legitimate hosts. These analyses identify open research questions and future research directions that could generate further work in this area.

Tao et al. [30] have designed, implemented, and extensively tested an architecture that allows LVM (live) migration to be controlled using policy. Migration is intended for maintenance, load balancing, or as a security mechanism called Moving Target Defence. The migration mechanism is easily configured via a configuration file and they proposed new policy-based architecture. They evaluate and analyze the system's performance in several scenarios on a local Mininet-based testbed. It is similar to ours in terms of policy-based migration but differs in that they only used it to distinguish the purpose of migration.

Huang et al. [31] noted that security is compromised due to immature auditing procedures of Docker images. To protect the security of host computers and local Docker containers from malicious Docker container attacks, it is necessary to detect potential threats of Docker images and discover risks when running Docker container instances on host computers. They provide a detailed analysis of the existing security mechanisms of Docker and the main threats that Docker users must face. As a result, they present threat detection techniques for Docker images and container instances and prove the effectiveness of the proposed detection framework through experimental results. The results are very similar to ours in terms of container security. The difference is that we aim to prevent further security risks due to migration in environments that use protected images through these mechanisms.

The above-mentioned studies proposed solutions for the quality of service and scheduling of containers in edge computing. In other words, these algorithms can automatically

control many containers, making it difficult for humans to determine whether a container can be placed or moved to a particular location. However, our proposed method can be used as a solution to this problem, that is allowing an automated decision whether a container can be migrated.

3. Data-Audit Control Mechanism

3.1. Data-Audit Control Mechanism for Virtual Machines

We have proposed and implemented a data-audit mechanism to migrate virtual machines in the cloud [32,33]. Virtual machine migration in the cloud is the physical transfer of virtual machines to another host machine. Virtual machine migration has become an indispensable technology for cloud computing, it is simplified and can be easily performed by those with the cloud's privileges. The problem of virtual machine migration is that the migration process is simplified and can be easily executed if the cloud's permissions are met. That is, unintentional migration of virtual machines to the wrong destination or the malicious migration of virtual machines can be carried out. This can lead to violations of company rules and national laws. This is a serious problem and a control mechanism is needed to determine whether migration is allowed, taking into account the authority of the cloud, the physical availability of the virtual machine, and also the data that the virtual machine contains.

To solve the problem of the inappropriate migration of data contained in virtual machines, we proposed a data-audit control mechanism based on policies (Figure 3). The virtual machine administrator writes a list of identifiers of countries and organizations that can be migrated based on the regulations attached to the data in the virtual machine in the REGULATION. <CountryCode> in the REGULATION indicates the countries where data movement is permitted based on regulations. <OrganizationCode> in the REGULATION indicates the organization that instantiates and uses the virtual machine. The host machine administrator describes the country location of the host machine in COUNTRY and the organization that manages the host machine in ORGANIZATION. When the migration is executed, the REGULATION of the virtual machine to be migrated is compared with the COUNTRY and ORGANIZATION of the host machine to be migrated. First, it checks whether <CountryCode> in the REGULATION contains the COUNTRY of the destination host machine. Similarly, it checks whether <OrganizationCode> in the REGULATION contains the ORGANIZATION of the destination host machine. If both checks pass, this signifies that the data acquired by the virtual machine can be migrated to the destination host machine and migration is executed. Migration execution uses an existing migration process. These mechanisms avoid unintentional data breaches caused by data migration under the terms and conditions granted by the various owners of the data, national laws and regulations, and organizational policies.

We implemented and evaluated these mechanisms in a cloud environment. We verified whether migrating virtual machines have complied with the policies and confirmed that the decision was made accurately. We also measured the execution time of live migration using our proposed mechanisms and showed no overhead cost. In this paper, we extend our proposed mechanism from virtual machines in the cloud to containers in edge computing environment.

3.2. Data-Audit Control Mechanism for Containers

The improper movement of data is a serious concern in containers and virtual machines migration. Therefore, we propose a policy-based data-audit control mechanism for container migration by extending our data-audit control mechanism for virtual machines. Unlike host machines, the edge server does not have abundant resources, requiring hardware requirements/attributes. Therefore, our proposed mechanism adds the regulation requirements/attributes and the hardware requirements/attributes to the data-auditing policies. The regulation requirements/attributes checks whether the data can be migrated in a software manner and the hardware requirements/attributes check whether the data

can be migrated in hardware. At the container, the container owner writes the container requirements based on the container's application and data conventions. At the edge server, the edge server owner describes the edge server attributes (Figure 4). When a container is to be migrated, the container requirements are compared with the attributes of the destination edge server to determine whether the migration is feasible. The migration shall be performed if the destination edge server satisfies the requirements at the container (Figure 5).

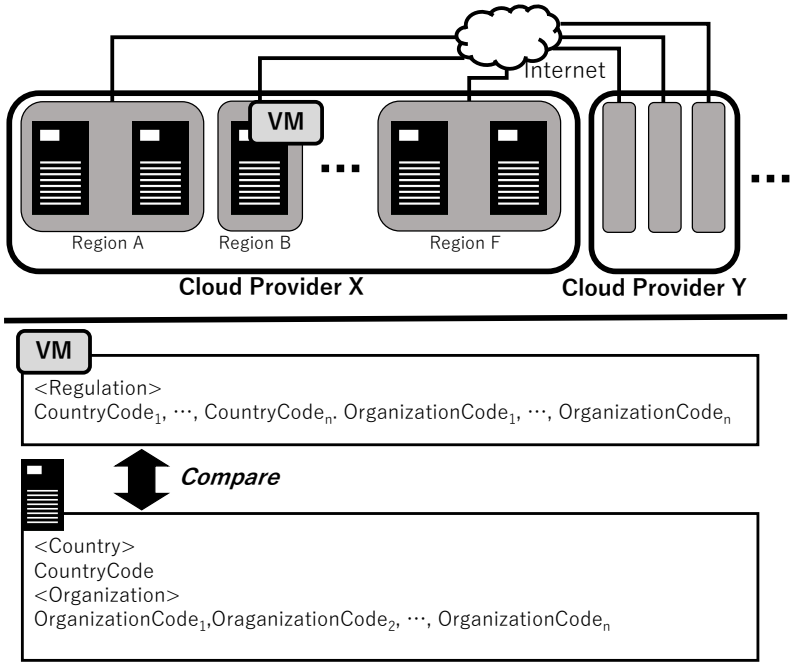

Figure 3. The proposed virtual machine migration control mechanism using policies.

In addition to specifying which edge server, there are two other methods: one is to specify the region where the edge server belongs and automatically select an edge server that satisfies the requirements. The other is to specify neither an edge nor a region but to automatically select an edge that satisfies the requirements. When migration is performed by specifying a region, the edge servers within the selected region are selected randomly to determine whether migration is possible. If the condition is not met, this means another edge server in the same region is using the same process. The process is repeated until the condition is satisfied or there are no more edge servers in the region. If neither an edge server nor a region is specified, the process is repeated until the condition is satisfied or there are no more edge servers in the region.

3.2.1. Data-Auditing Policy for Container Requirements

The container owner describes the container requirements in the data-auditing policy that determine whether the container can be migrated (Table 1). In regulation requirements: the cDAl describes the countries in which the container can be placed. The cDAo describes the platforms allowed to deploy the container based on the conventions of the organization that manages the container. In hardware requirements, the cHRh describes the manufacturer name of the allowed edge servers. The cDAo describes the platform allowed to deploy the container based on the conventions of the organization that manages the container. The cHRh is a list of allowed edge servers. The cHRt describes the tools used by the container. The cHRcu describes the CPU utilization threshold of the edge server to be migrated. The cHRmu describes the memory utilization threshold of the edge server to be migrated. The cHRc describes the memory utilization threshold of the edge server to be migrated and it also describes the minimum CPU frequency threshold for the destination edge server. The cHRm describes the minimum memory capacity threshold for the destination edge server.

The cHRgpu describes whether the destination edge server must support AI processing. The cHRaf describes the AI framework required at the destination edge server.

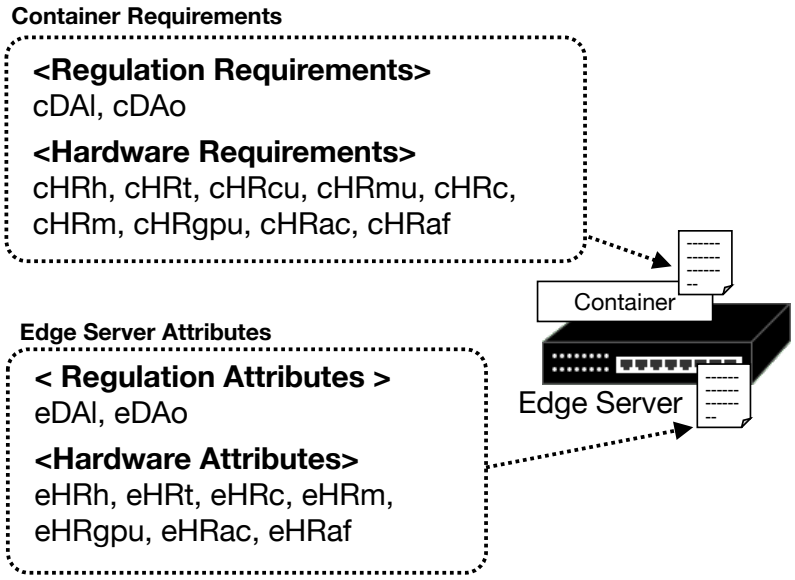

Figure 4. Data-auditing policy of container and edge server.

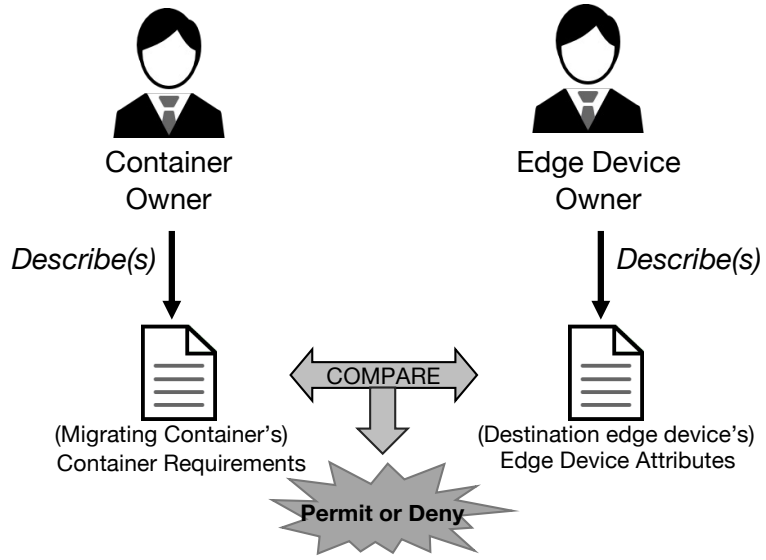

Figure 5. The process of container migration with data-audit control mechanism.

Table 1. Container requirements in the data-auditing policy.

Element	Description
cDAl	Countries or regions that are allowed.
cDAo	Administrative organizations that are allowed.
cHRh	Name of edge computing device manufacturers that are allowed.
cHRt	Virtualization tools to be used.
cHRcu	CPU utilization at the destination edge computing device (%).
cHRmu	Memory utilization at the destination edge computing device (%).
cHRc	Min CPU frequency at the destination edge computing device (GHz).
cHRm	Min memory capacity at the destination edge computing device (GB).
cHRgpu	AI support availability the destination edge computing device.
cHRac	AI accessories required the destination edge computing device.
cHRaf	AI framework required the destination edge computing.

3.2.2. Data-Auditing Policy for Edge Server Attributes

An edge server owner describes the edge server attributes in the data-auditing policy, which is necessary to determine whether to migrate into the edge server (Table 2). eDAl describes the country where the container is located. eDAo describes the organization that manages the container. eHRh describes the manufacturer name of the edge server. eHRt describes the policy of the edge server. eDAl describes the country where the container is located. eDAo describes the organization that manages the container. eHRh describes the manufacturer of the edge server. eHRt describes the tools available. eHRc describes the CPU frequency of the edge server. eHRm describes the memory capacity of the edge server. eHRc describes the edge server's CPU frequency. eHRm describes the edge server's memory capacity threshold. eHRgpu describes whether AI processing is supported. eHRac describes the available AI accessories. eHRaf describes the available AI framework. eHRt describes the available tools.

Table 2. Edge server attributes in the data-auditing policy.

Element	Description
eDAl	Country of placement.
eDAo	Managing organization.
eHRh	Edge computing device manufacturer name.
eHRt	Available virtualization tools.
eHRc	CPU frequency (GHz).
eHRm	Memory capacity (GB).
eHRgpu	AI support.
eHRac	Available AI accessors.
eHRaf	Available AI frameworks.

3.2.3. Comparison of Policies

For example, if the container described in Table 3 attempts to migrate to the edge server with attributes in Table 4, the migration is not allowed and will not be executed.

Table 3. Container requirements.

cDAl	cDAo	cHRh	cHRt	cHRcu	cHRmu	cHRc	cHRm	cHRgpu	cHRac	cHRaf
jp	camA	devA	lxc	40	40	2.2	2	-	-	-

Table 4. Edge server attributes (CPU usage = 10%, Memory usage = 20%).

eDAl	eDAo	eHRh	eHRt	eHRc	eHRm	eHRgpu	eHRac	eHRaf
us	camA	devA	lxc	2.6	8	-	-	-

The cDAl of the container requirements does not satisfy the condition because it is not equal to the eDAl of the edge server attributes. The container requirements' cDAo satisfies the eDAo of the edge server attributes. The container requirements' cHRh satisfies the conditions because it equals the eHRh of the edge server attributes. The cHRt of the container requirements equals the eHRt of the edge server attributes, thus satisfying the condition. The container requirements' cHRcu is greater than or equal to 10, the value of the edge server attributes' CPU usage, so the condition is satisfied. The container requirements' cHRmu is greater than or equal to 20, which is the edge server's memory usage, so the condition is satisfied. The cHRc of the container requirements is greater than or equal to the eHRc of the edge server attributes, thus satisfying the condition. The cHRm of the container requirements is greater than or equal to the eHRm of the edge server attributes, thus satisfying the condition. cHRgpu, cHRac, and cHRaf of the container requirements are not considered because they do not fulfill the condition. Therefore, the migration is not

allowed and the container migration is not performed because the items that do not satisfy the condition are more than 1.

3.3. System Design of the Data-Audit Control Mechanism

This section described the design of the system that implements the policy-based data-audit control mechanism. By adding a new data-audit cloud unit to the existing edge computing environment of edge servers and containers, the data-audit control mechanism can be utilized during container migration, shown in Figure 6. The data-audit cloud unit includes an Edge/Container Management Server, a Data-Audit Execution Server, and a UI Server. The Edge/Container Management Server runs the Monitor Module for monitoring the edge servers and containers and the Control Module for operating the edge servers and containers. The Data-Audit Execution Server runs the core mechanism for the data audit and other processes, such as the approval or disapproval of migration, are performed by this server. The Data-Audit Execution Server contains a database that stores the edge server Policy and Container Policy used for the data audit. The UI Server allows the edge server and container owners to use the data-audit system visually. The Execution Server communicates with the Data-Audit Server via the REST API.

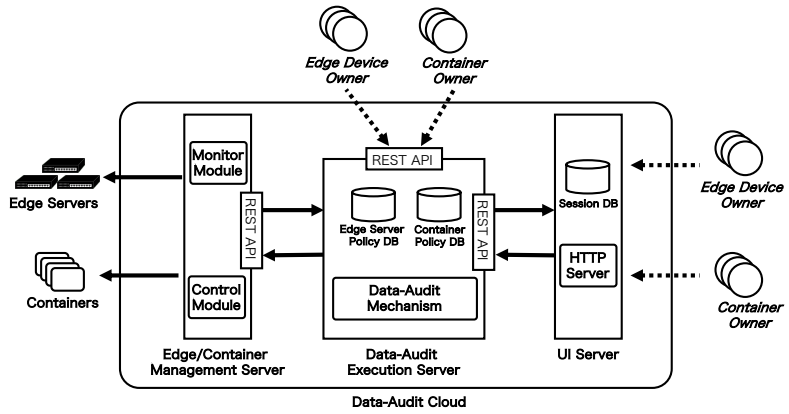

Figure 6. Design a system that implements a policy-based data-audit control mechanism.

3.4. Data-Audit Control Flow

Figure 7 shows the flow of registering information on existing edge servers and containers to the data-audit control system using the UI Server. The edge server owners register their edge servers with the UI Server. The registered information is stored as a policy in the edge server Policy database in the Data-Audit Execution Server. The container owners register their containers and policies with the UI Server. The registered information is stored in the Container Policy database in the Data-Audit Execution Server and the Edge/Container Management Server monitors the registered edge servers and containers every 10 min, as shown in Figure 8. For the edge servers, it obtains the CPU and memory utilization and survival checks. For containers, only survival checks are performed. The flow of the data audit during migration is shown in Figure 9. The container owner sends a migration request along with the target edge server to the UI Server, which sends a request to the Data-Audit Execution Server for the target edge server and the migration. The Execution Server requests information about the target edge server from the Edge/Container Management Server and receives CPU and memory utilization. The data-audit mechanism uses these values. If the data-audit results do not permit migration, a failure notice is returned to the container owner. If the migration is allowed, the migration is executed via the Edge/Container Management Server. A success notice is returned to the container owner.

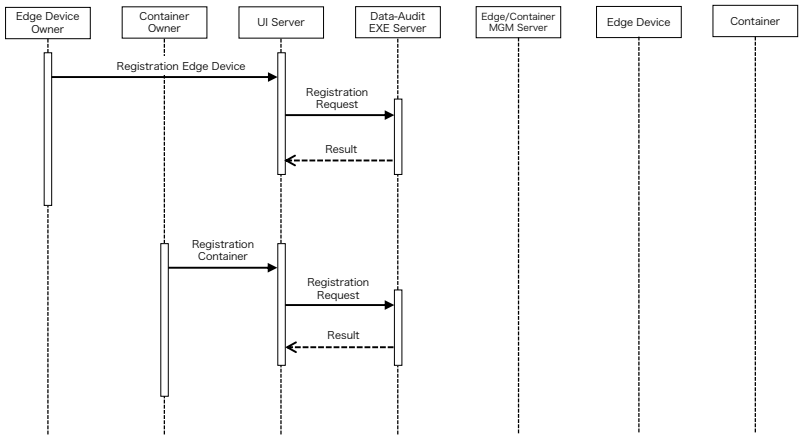

Figure 7. Registration process for container and edge.

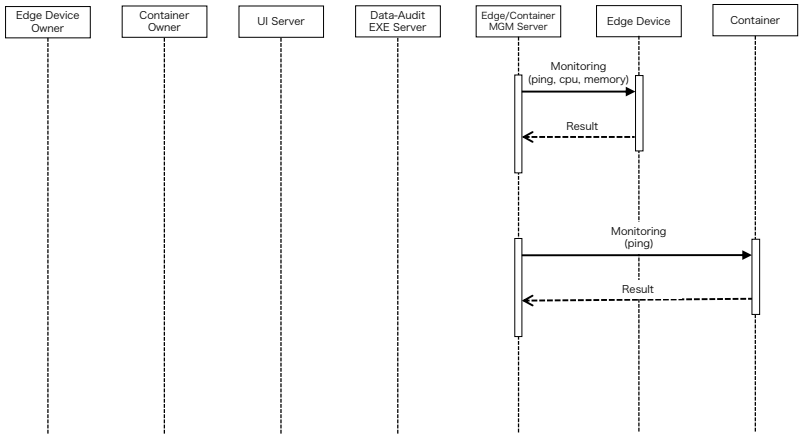

Figure 8. Monitoring process for registered container and edge.

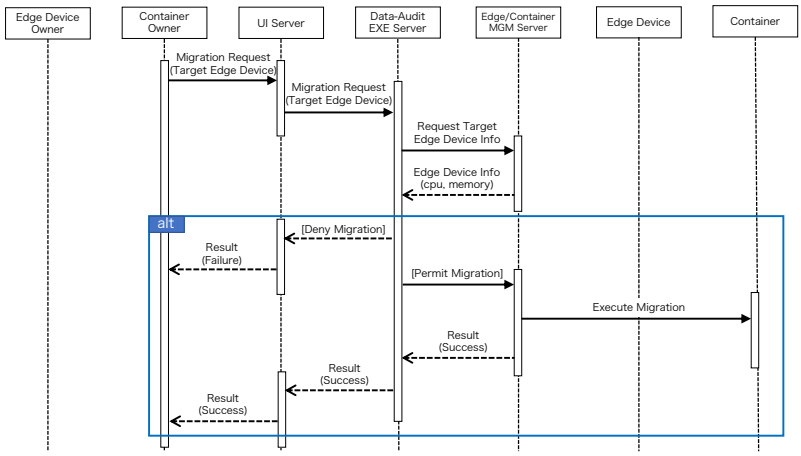

Figure 9. Migration with data-audit process.

4. Implementation of the Data-Audit Control System

4.1. Implementation of WebUI

There are two ways to use the data-audit control system: by directly using the Data-Audit Execution Server via the REST API or by accessing the UI Server via a Web browser. This section introduces a method for data-audit migration using the UI Server, which has two attributes: edge server owner and container owner.

Using a data-audit control system, the edge server owners can register the edge servers they manage and include them in container migrations. First, the user registers as the owner of the edge server. After that, the user logs in from the login page (Figure 10) and

moves to the top page of the edge server owner (Figure 11). A list of registered edge servers on the top page can be viewed on the top page. The registered edge servers can be confirmed by checking the registered name, IP address, several active containers, and health status. Clicking on the icon of an edge server that someone wishes to view from the list of edge servers allows one to check the details of the edge server (Figure 12). A new edge server can be registered from the top page. The registration screen is shown in Figure 13.

Container owners can register their own containers to check its status in real-time and perform migration between the edge servers in which they were registered and permitted to manage. First, a user registers as a container owner. After that, the user logs in from the login page and moves to the top page of the container owner. On the top page, a list of permitted and registered edge servers and containers managed by the container owner can be viewed (Figure 14). The edge server registration from a share key is shown in Figure 15 and the container registration in Figure 16. Migration execution can be performed by clicking the buttons. The authentication details of each can be viewed by clicking on the edge server and container icons (Figure 17). To execute the migration, select the container and the edge server to be migrated and start the migration process. During the migration process, a data audit is performed and the migration is executed if the conditions are met. After the migration is executed, the information in the list of containers in the migration destination is updated.

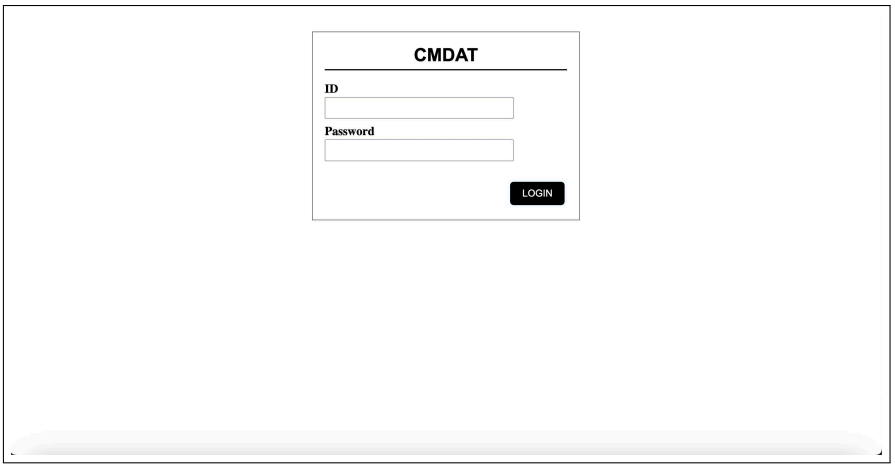

Figure 10. Login page.

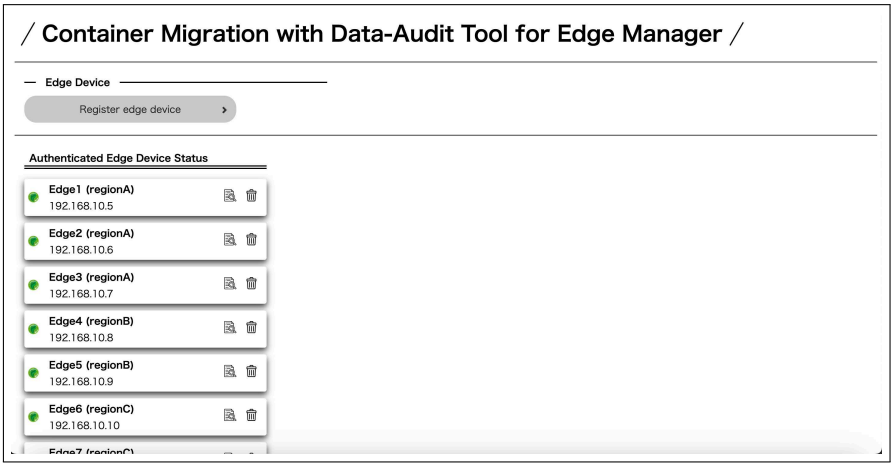

Figure 11. Top page for edge owner.

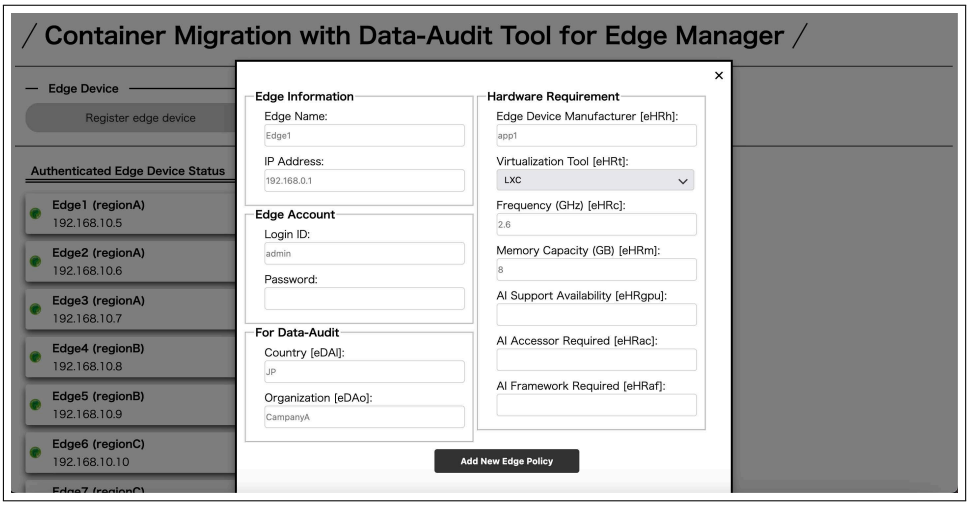

Figure 12. Edge servers registration window.

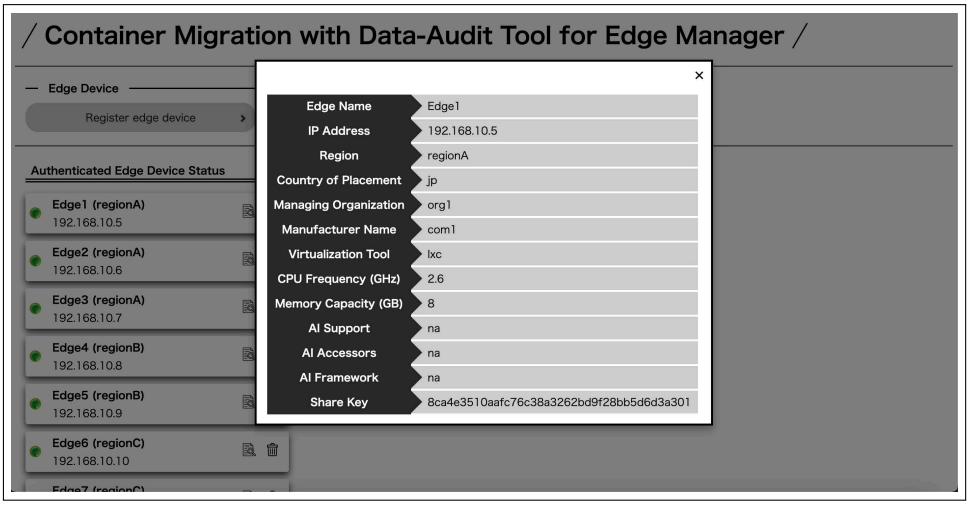

Figure 13. Detail window of own edge server.

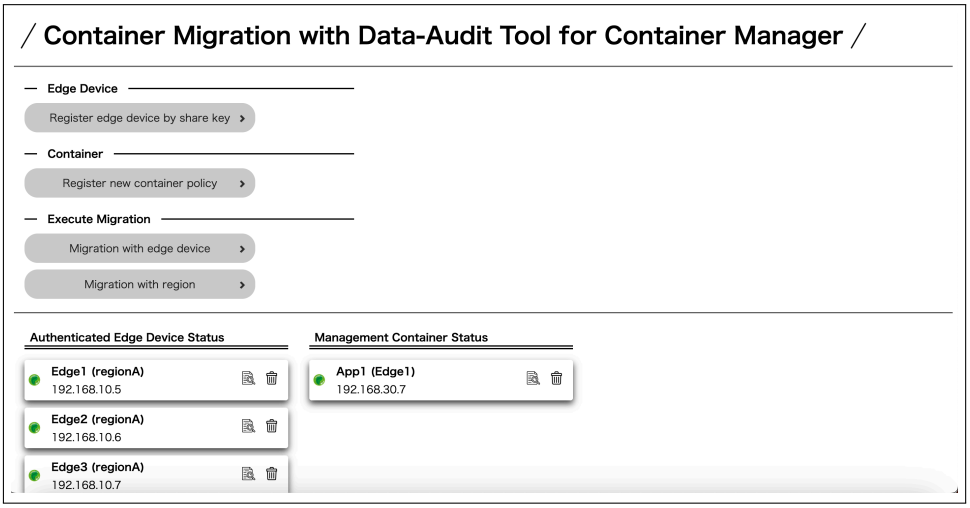

Figure 14. Top page of container owner.

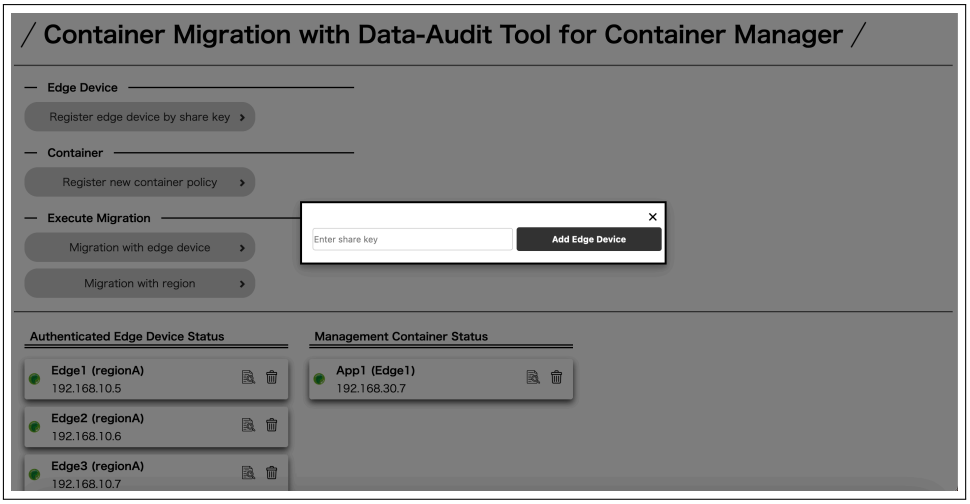

Figure 15. New edge server registration window from share key.

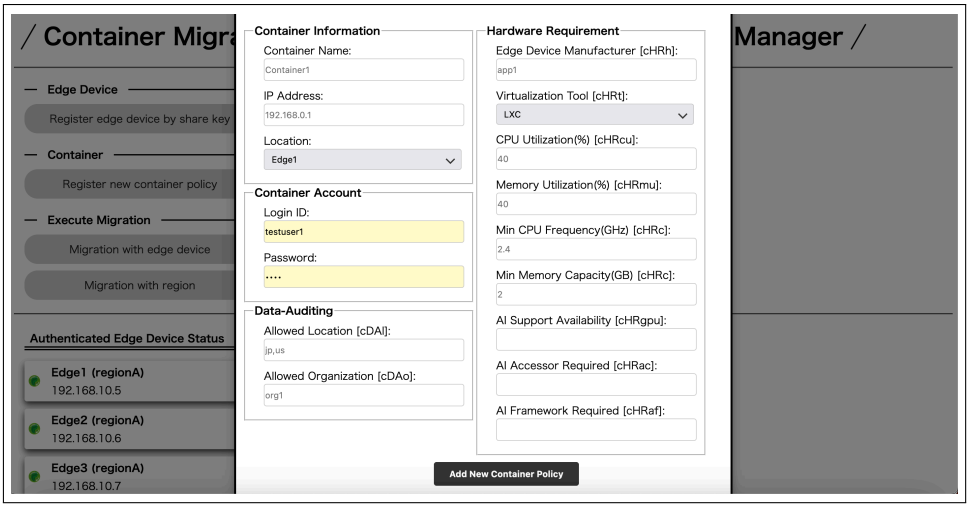

Figure 16. New container registration window.

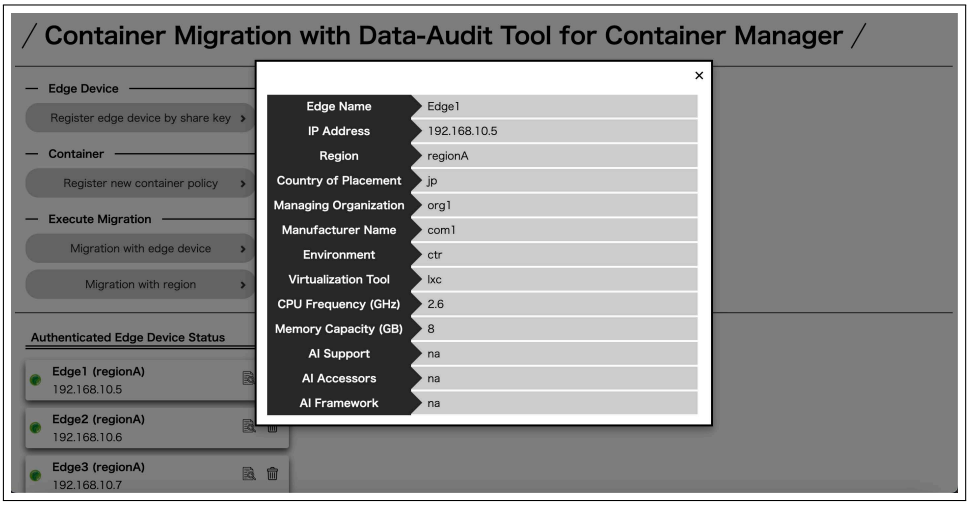

Figure 17. Detail window of authenticated edge server.

4.2. Verification Environment

We implemented an edge computing environment consisting of multiple edge servers divided into multiple regions. All edge servers are assumed to be running LXC and running container-generated applications. The CRIUs are deployed at all edge servers so containers

can be migrated between the edge servers. Furthermore, the components to realize the proposed data-auditing policy are installed to all the edge servers. The implementation environment is shown in Figure 18.

The implemented edge computing environment consists of eight edge servers and four regions. Edge 1 is running a container. These edge servers are virtual machines on an ESXi-6.5.0 high-performance server. The server consists of 20 CPUs, Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60 GHz, 128 GB memory, and 3.45 TB HDD storage. All the edge servers are configured with 2 vCPUs, 8 GB memory, 80 GB storage, and Ubuntu 22.04 LTS OS. LXC and CRIU were installed and ready to run the containers.

The container requirements of the App1. container running on Edge 1 are shown in Table 5. The edge server attributes are shown in Table 6.

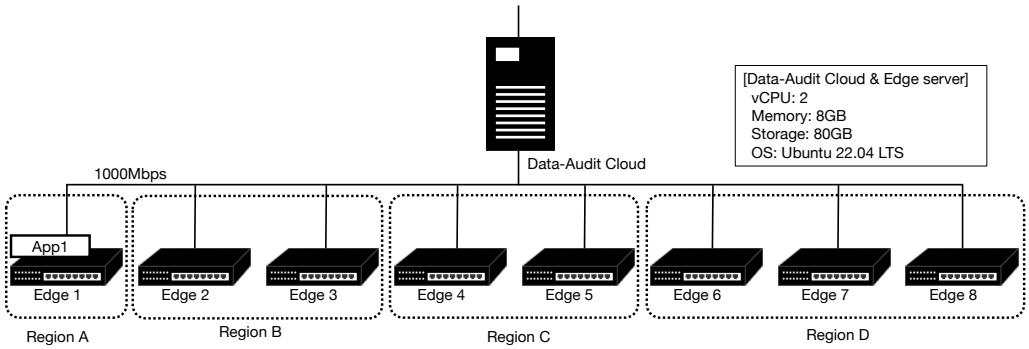

Figure 18. Implementation environment.

Table 5. App1's container requirements. (* indicates that everything is permitted.)

cDAl	cDAo	cHRh	cHRt	cHRcu	cHRmu	cHRc	cHRm	cHRgpu	cHRac	cHRaf
jp,us,fr	-caB	*	lcx	60	50	1.6	4	*	*	*

Table 6. Edge server attributes (CPU usage = 40%, Memory usage =30%).

	eDAl	eDAo	eHRh	eHRt	eHRc	eHRm	eHRgpu	eHRac	eHRaf
Edge1	jp	caA	deA	lxc	2.6	8	-	-	-
Edge2	de	caA	deB	lxc	2.6	8	-	-	-
Edge3	fr	caA	deC	lxc	2.6	8	-	-	-
Edge4	us	caB	deD	lxc	2.6	8	-	-	-
Edge5	us	caC	deE	lxc	2.6	8	-	-	-
Edge6	kr	caD	deF	lxc	2.6	8	-	-	-
Edge7	jp	caE	deG	lxc	2.6	8	-	-	-
Edge8	jp	caE	deH	lxc	2.6	8	-	-	-

5. Verification

This section examines three cases in an experiment environment that implements a data-audit control system. Case 1 is where the migration is performed by specifying the edge server to be migrated. In Case 2, the migration is performed by specifying the region to which the edge server belongs. In Case 3, we examine the case where neither the edge server nor the region is specified.

5.1. Case 1: Specify Edge Server

This section examines the execution time and its breakdown when migration is performed by specifying edge servers. The execution times for migrating containers running on from Edge 1 to Edge 2–8 are shown in Figure 19. In all cases, the average execution time of 10 runs of each migration is used. Edge 2, 4, and 6 do not satisfy the condition, so the migration is not executed. The migration is performed to the other edge servers because they satisfy the conditions. To Edge 3, the time required for the data audit was 0.148 ms,

while the migration required 14.453 ms. The time required for the data audit was 0.9% of the total time and the impact on migration was considered very small.

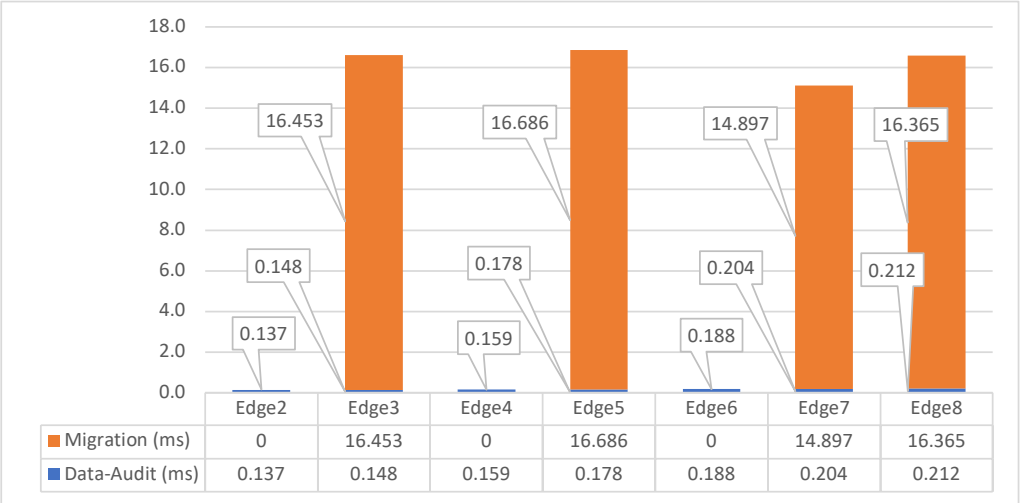

Figure 19. Execution time breakdown when migrating to from Edge 1 to Edge 2–8.

5.2. Case 2: Specify Region

This section examines the execution time and its breakdown when the migration was executed by specifying a region. Table 7 shows the execution times for migrating containers running on from Edge 1 to Region B, C, and D, respectively. In Region B, the migration was performed to Edge 3. In Region C, the migration was performed to Edge 5. In Region D, the migration was performed to Edge 7 or Edge 8. The average execution time was 16.175 ms. In Region B, the data audit required 0.062 ms, while the migration required 16.003 ms. The time required for the data audit was 0.39% of the total time and the impact of the data audit on the migration was considered very small.

Table 7. Execution time breakdown when migrating containers running on from Edge 1 to Region B–D.

	to Region B	**to Region C**	**to Region D**
1st 2nd	Edge2 -> Deny Edge3 -> Permit	Edge4 -> Deny Edge5 -> Permit	Edge8 -> Permit
Data audit (ms) Migration (ms)	0.062 15.941	0.075 15.74	0.042 16.664
Total (ms)	16.003	15.815	16.706

5.3. Case 3: Specify Nothing

This section examines the execution time and its breakdown when migration is performed without specifying either the edge server or the region. Table 8 shows the execution time and destination edge servers for five migrations. Edges 2, 4, and 6 did not satisfy the conditions, so the data-audit system did not execute the migration. The migrations were performed to the other edge servers because they satisfied the conditions. The average execution time was 15.193 ms. The minimum time required for the data audit was 0.026ms and the maximum was 0.064 ms. The time required for the data audit was between 0.18% and 0.46% of the total time and the impact on migration was considered very small.

Table 8. The execution time and destination edges for five migrations from Edge 1.

	#1	#2	#3	#4	#5
1st 2nd	Edge5 -> Permit	Edge6 -> Deny Edge5 -> Permit	Edge2 -> Deny Edge7 -> Permit	Edge7 -> Permit	Edge4 -> Deny Edge5 -> Permit
Data-audit (ms) Migration (ms)	0.033 13.895	0.064 16.649	0.054 16.742	0.026 14.507	0.064 13.9
Total (ms)	13.960	16.713	16.796	14.533	13.964

6. Conclusions

In recent years, we have focused our work on containers and their migration, whose demand is rapidly growing. We foresaw that the same data-audit problems we face with virtual machine migration will also occur in containers. Therefore, we proposed a data-audit control mechanism for container migration, extending the control mechanism we have already proposed for virtual machine migration. The proposed data-audit control system checks whether data can be migrated in compliance with the regulations and whether data can be migrated in compliance with the hardware requirements. When a container is to be migrated, it is compared with the data-auditing policy of the edge server attributes where the migration is scheduled to occur to determine whether the migration is allowed or not, thereby avoiding data compromise caused by the migration. We have built and verified a data-audit system that implements the proposed mechanism in an edge computing environment. We found that, in three cases, the addition of the proposed data-audit system had a minimal impact on the migration time, indicating that the system is practical enough. The WebUI was introduced in this implementation to simplify the description of the attributes and policies. However, since the possibility of erroneous entries cannot be eliminated, we are planning to implement a mechanism to support such entries.

In the future, we intend to conduct verification not in a very compact and short-range environment, as in the current environment, but in a network that mimics an existing wide-area network or on an existing wide-area network. Moreover, we aim to ensure the solution is compatible with current container orchestration/management services by open sourcing the solution and ensuring it is available as an API.

Author Contributions: Conceptualization, T.U. and M.H.; methodology, T.U. and M.H.; validation, T.U, B.A., T.S. and M.H.; investigation, T.U.; resources, T.U. and T.S.; data curation, T.U.; writing—original draft preparation, T.U.; writing—review and editing, T.U., B.A., T.S. and M.H.; visualization, T.U.; supervision, T.U.; project administration, T.U.; funding acquisition, T.U. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JSPS KAKENHI Grant Number JP20K19778.

Data Availability Statement: Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Atzori, L.; Iera, A.; Morabito G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Tan, L.; Wang, N. Future internet: The Internet of Things. In Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China, 20–22 August 2010; pp. V5-376–V5-380.
3. Dhananjay, S.; Tripathi, G.; Jara, A.J. A survey of Internet-of- Things: Future vision architecture challenges and services. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Republic of Korea, 6–8 March 2014; pp. 287–292.
4. KVM. Available online: https://www.linux-kvm.org/ (accessed on 26 October 2022).
5. Xen Project. Available online: https://xenproject.org/ (accessed on 26 October 2022).
6. Shetty, J.; Anala, M.R.; Shobha, G. A Survey on Techniques of Secure Live Migration of Virtual Machine. *Int. J. Comput. Appl.* **2012**, *39*, 34–39. [CrossRef]

7. Aiash, M.; Mapp, G.; Gemikonakli, O. Secure Live Virtual Machines Migration: Issues and Solutions. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; pp. 160–165.
8. Upadhyay, A.; Lakkadwala, P. Secure live migration of VM's in Cloud Computing: A survey, Reliability. In Proceedings of the 3rd International Conference on Reliability, Infocom Technologies and Optimization, Noida, India, 8–10 October 2014; pp. 1–4.
9. Rathod, N.; Chauhan, S. Survey: Secure Live VM Migration In Public Cloud. *Int. J. Sci. Res. Dev.* **2015**, *2*, 271–274.
10. Flores H.; Tran V.; Tang, B. PAM & PAL: Policy-Aware Virtual Machine Migration and Placement in Dynamic Cloud Data Centers. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 2549–2558.
11. Jena, S.; Sahu, L.K.; Mishra, S.K.; Sahoo, B. VM Consolidation based on Overload Detection and VM Selection Policy. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 252–256.
12. Gutierrez-Garcia, J. O.; Ramirez-Nafarrate, A. Policy-Based Agents for Virtual Machine Migration in Cloud Data Centers. In Proceedings of the 2013 IEEE International Conference on Services Computing, Santa Clara, CA, USA, 28 June–3 July 2013; pp. 603–610.
13. Koto, A.; Kono, K.; Yamada, K. A Guideline for Selecting Live Migration Policies and Implementations in Clouds. In Proceedings of the 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 15–18 December 2014; pp. 226–233.
14. Cui, L.; Tso, F. P.; Pezaros, D.P.; Jia, W. PLAN: A Policy-Aware VM Management Scheme for Cloud Data Centres. In Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Limassol, Cyprus, 7–10 December 2015; pp. 142–151.
15. Papadopoulos, A.V.; Maggio, M. Virtual Machine Migration in Cloud Infrastructures: Problem Formalization and Policies Proposal. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 6698–6705.
16. Singh, P.; Gupta, P.; Jyoti, K. Energy Aware VM Consolidation Using Dynamic Threshold in Cloud Computing. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019; pp. 1098–1102.
17. Ibrahim, A.; Noshy, M.; Ali, H.A.; Badawy, M. PAPSO: A Power-Aware VM Placement Technique Based on Particle Swarm Optimization. *IEEE Access* **2020**, *8*, 81747–81764. [CrossRef]
18. Shirazi, N.; Simpson, S.; Marnerides, A.K.; Watson, M.; Mauthe, A.; Hutchison, D. Assessing the impact of intra-cloud live migration on anomaly detection. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 8–10 October 2014; pp. 52–57.
19. Kantarci, B.; Mouftah, H.T. Resilient design of a cloud system over an optical backbone, *IEEE Netw.* **2015**, *29*, 80–87. [CrossRef]
20. Fu, X.; Zhang, C.; Chen, J.; Zhang, L.; Qiao, L. Network Traffic based Virtual Machine Migration in Cloud Computing Environment. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 818–821.
21. Yazidi, A.; Ung, F.; Haugerud, H.; Begnum, K. Affinity Aware-Scheduling of Live Migration of Virtual Machines Under Maintenance Scenarios. In Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; pp. 1–4.
22. Deshpande, L.; Liu, K. Edge computing embedded platform with container migration. In Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017; pp. 1–6.
23. Puliafito, C.; Vallati, C.; Mingozzi, E.; Merlino, G.; Longo, F.; Puliafito, A. Container Migration in the Fog: A Performance Evaluation. *Sensors* **2019**, *19*, 1488. [CrossRef]
24. Karhula, P.; Janak, J.; Schulzrinne, H. Checkpointing and Migration of IoT Edge Functions. In Proceedings of the 2nd International Workshop on Edge Sys- tems, Analytics and Networking (EdgeSys'19), Dresden, Germany, 5 March 2019; pp. 60–65.
25. Home—Docker. Available online: https://www.docker.com/ (accessed on 26 October 2022).
26. Linux Containers. Available online: https://linuxcontainers.org/ (accessed on 26 October 2022).
27. Open-Source Container-Based Virtualization for Linux. Available online: https://openvz.org/ (accessed on 26 October 2022).
28. Red Hat OpenShift Makes Container Orchestration Easier. Available online: https://www.redhat.com/en/technologies/cloud-computing/openshift (accessed on 26 October 2022).
29. Sultan, S.; Ahmad, I.; Dimitriou, T. Container Security: Issues, Challenges, and the Road Ahead. *IEEE Access* **2019**, *7*, 52976–52996. [CrossRef]
30. Tao, X.; Esposito, F.; Sacco, A.; Marchetto, G. A Policy-Based Architecture for Container Migration in Software Defined Infrastructures. In Proceedings of the 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 24–28 June 2019; pp. 198–202.

31. Huang, D.; Cui, H.; Wen, S.; Huang, C. Security Analysis and Threats Detection Techniques on Docker Container. In Proceedings of the 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 6–9 December 2019; pp. 1214–1220.
32. Uchibayashi, T.; Hashi, Y.; Hidano, S.; Kiyomoto, S.; Apduhan, B.O.; Abe, T.; Suganuma, T.; Hiji, M. A Control Mechanism for Live Migration with Data Regulations Preservation. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Athens, Greece, 3–6 July 2017; pp. 509–522.
33. Uchibayashi, T.; Apduhan, B O.; Shiratori, N.; Suganuma, T.; Hiji, M. Policy Management Technique Using Blockchain for Cloud VM Migration. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; pp. 360–362.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.