

A Cloud VM Migration Control Mechanism Using Blockchain

Toshihiro Uchibayashi[1(✉)], Bernady Apduhan[2(✉)], Takuo Suganuma[3(✉)], and Masahiro Hiji[3(✉)]

[1] Kyushu University, Fukuoka, Japan
uchibayashi.toshihiro.143@m.kyushu-u.ac.jp
[2] Kyushu Sangyo University, Fukuoka, Japan
bob@is.kyusan-u.ac.jp
[3] Tohoku University, Sendai, Japan
{suganuma,hiji}@tohoku.ac.jp

Abstract. In the current cloud, VM migration that moves VMs between physical host machines is indispensable. For cloud providers, before shutting down the physical host machines for maintenance, migration is used to temporarily save VMs to other physical host machine. For the cloud user, migration is used to move a VM to a location which is geographically close to the end user. These VM migrations can be performed very easily and are only limited by the scope of the VM administrator's contract. However, the problem lies on the permission of the data in VM. In recent years, with the widespread use of IoT, various types of data can be stored in cloud's VMs through web services. The huge amount of data collected by IoT devices requires close attention to manage because it could be very closely related to the information of an individual. However, there is no mechanism for checking data permission in VM during VM migration, and there is concern that inappropriate data movement may occur. This includes the unintended risky movement of inappropriate data which could be malicious data. Therefore, we proposed a mechanism to ensure compliance with the conditions granted by the data owner, the country regulations, and the organization regulations during VM migration. By constructing the proposed mechanism in blockchain, we can prevent malicious tampering and thus enable robust VM migration control.

Keywords: Cloud computing · VM migration · Audit system · Privacy policy · Data protection · Blockchain

1 Introduction

With the widespread use of the Internet of Things (IoT), it has become common to store in the cloud a huge amount of data collected from various sensors that measures the health of an individual and the environment linked to that particular individual [1–3]. This huge amount of data collected by IoT devices

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12252, pp. 221–235, 2020.
https://doi.org/10.1007/978-3-030-58811-3_16

reveals not only the individual's personal values but also multiple data relations that likewise reveal useful means in predicting personal preference and behavior, intervention in behavior, etc. This data analysis is performed by the service provider. These data are closely related to individuals and require careful management. Appropriate management and use in compliance with the regulations of the country or region like General Data Protection Regulation (GDPR) [4], as well as the regulations of the organization, and the conditions granted by the data owner are essentials.

Moreover, not only IoT, but also services using a cloud environment are generally used to operate applications and services via networks, and the operations in a cloud environment are already pervasive in our current world. As the demand for cloud services increases, businesses providing cloud environments is explosively increasing and are developing various services. In particular, operators providing large-scale cloud services such as AWS [5] and Azure [6] are deploying services on a global scale, and cloud users can select a data center or service that suits them. In recent years, by combining serverless cloud services, various web services can be deployed without building servers. However, serverless environments have various limitations, and traditional methods of building servers on VMs are indispensable. Among them, Infrastructure as a Service (IaaS), a cloud service that provides Virtual Machine (VM), is widely used because it can be used for general purposes. Businesses that provide IaaS are widely provided not only by large-scale businesses such as AWS, but also by small and medium-sized businesses.

What needs special attention when using the IaaS cloud is the inappropriate data movement. In the current cloud, using VM migration to move between physical host machines is indispensable. For cloud service providers, before stopping the physical host machine for maintenance, disaster recovery, etc., migration is used to temporarily store applications in VMs and move to other physical host machines. For cloud users, migration is used to move VM to a location geographically close to the end user. These VM migrations can be performed very easily and are only limited by the scope of the VM administrator's contract. Therefore, the problem is the permission of data in VM.

What needs special attention is the unintended movement of inappropriate data. VM is deployed and running on host machines in the cloud environment. It is easy for VM to migrate between host machines, and the administrator of VM can change the host machine on which VM runs as needed [7]. There is no problem if the host machines are all in the same place, but if the host machines are installed in different countries, it may cause violations of regulations and organizational rules. However, there is no mechanism yet for checking the permission of data in VM at VM migration, and there is concern that inappropriate data movement may occur. This includes the risks of unintended movement of inappropriate data which may include the movement of malicious data. Therefore, we propose a mechanism to ensure compliance with the conditions granted by the data owner and the regulations of the country, regulatory, and organization during VM migration. By constructing the proposed mechanism in blockchain, we can prevent malicious tampering and enable robust VM migration control.

In this paper, after demonstrating the mechanism of VM migration control, we implement and evaluate the proposed mechanism with blockchain to the existing cloud environment. Section 2 introduces related work and explain the difference from this research. Section 3 shows VM migration and its problems. Section 4 proposes a policy control mechanism using blockchain. Section 5 implements the proposed mechanism. Section 6 evaluates the proposed mechanism using the implemented environment, and Sect. 7 is the conclusion.

2 Related Work

The research on migration security for cloud environment [8–11] discusses security risks. They investigate attacks against live migration of VM and introduce research and examples for securing live migration. In conclusion, in current systems, there exists no integrated approach to address trust establishment, confidentiality and integrity of migration data, authentication and approval of migration operations required for secure migration. A comprehensive framework that addresses these security aspects of live migration of VM is desired. This paper falls under the certification and approval of migration operation that is necessary for safe migration. By applying the mechanism proposed in this paper to the existing cloud, it is possible to perform safe migration.

The policy control using role-based for cloud infrastructure operation [12–15], assumes all hardware in the cloud environment is safe, and adds the ability to control the user to safely perform the migration. This prevents information leakage due to the live migration of unauthorized users. In addition, load balancing policies are used to effectively balance the workload of the entire data center.

Likewise, the research efforts to perform data protection of IoT [16–21] tries to ensure the safety of data collected by IoT using blockchain technology.

Here, studies on migration control and data protection are being conducted, and security warnings for migration, policy control, and securing of data safety by blockchain are desired. However, there exists no cloud mechanism for data migration. Therefore, we propose a secure mechanism to prevent unintended data movement and inappropriate data movement triggered by malicious tampering.

In the model of storing and delivering data on the Web [22], it propose the use of IPFS's distributed file storage system and the integrity retention characteristics of the blockchain. In the P2P file system called IPFS, which enables files to be stored in a distributed system, by storing files on a distributed network, it is possible to search packets from multiple sources, thus bandwidth savings. Security is ensured by placing the time stamp on the data rather than putting it on the chain itself.

3 VM Migration

There are two types of VM migration: live migration (hot migration) and cold migration (offline migration). Live migration can be executed without stopping

the running VM, so it can be moved live without stopping the service in the VM. The entire memory image on the VM running on the physical host machine is transferred to another physical host machine, and does not stop or freeze the operation of the OS, application software, network connection, etc., while the migration is carried out. Strictly speaking, a momentary interruption in a millisecond occurs at the switching moment. However, there is no disconnection of the network session, etc., and the VM user does not know that the movement has been performed. Major hypervisors such as KVM [23] and XEN [24] are already supported, and are already in practical use. In cold migration, the VM is shut down and then migrated to another host machine. The VM running on a physical host machine is temporarily stopped, and the memory image is transferred to another physical host machine via storage, etc. The operation is then resumed on the migration destination host machine. The operating status of the software is inherited without shutting down or restarting the OS.

VM migration is an indispensable technology for cloud operation. However, challenges arise to simplify the migration execution and to be easily performed whenever the cloud authorization is satisfied. When performing a migration with GCEE [25] or OpenStack [26], it can execute with one command line or one button. This only determines the cloud's authority and whether it can be physically migrated, and does not take into account the VM's internal data at all. Therefore, the problem is the unintended move destination mistake of VM and the malicious move of VM (Fig. 1). For example, suppose that VM contains personal data that cannot cross the country. And, according to internal regulations, this personal data can be stored only in Region A and Region B. There is no problem when migrating to a host machine of a region B in County X. However, if a VM is migrated to Region C by the VM manager's mistake, it will be a violation of the company policy. Also, if a malicious person migrates a VM to Region D in Country Y, it may violate the laws of the Country X. This is a very serious problem, and it is necessary to have a control mechanism that decides whether to migrate or not, considering not only the authority and physical availability of the cloud, but also the data contained in the VM.

4 Proposal of Control Mechanism

4.1 Blockchain Technique

Blockchain is a technology that records transaction data in units of blocks on nodes and records them in a distributed manner and manages the same block information. It is also called distributed management ledger technology. It is called a blockchain because the blocks are connected chronologically, like a chain. Blockchain consist of P2P networks, consensus algorithms, electronic signature and hash functions, and smart contracts. A P2P network provides a method of connecting computers for the same purpose to form a network. The consensus algorithm provides an algorithm for consensus building on distributed networks. Digital signatures and hash functions provide security mechanisms, such as a mechanism that guarantees the legitimacy of the person issuing a transaction. A

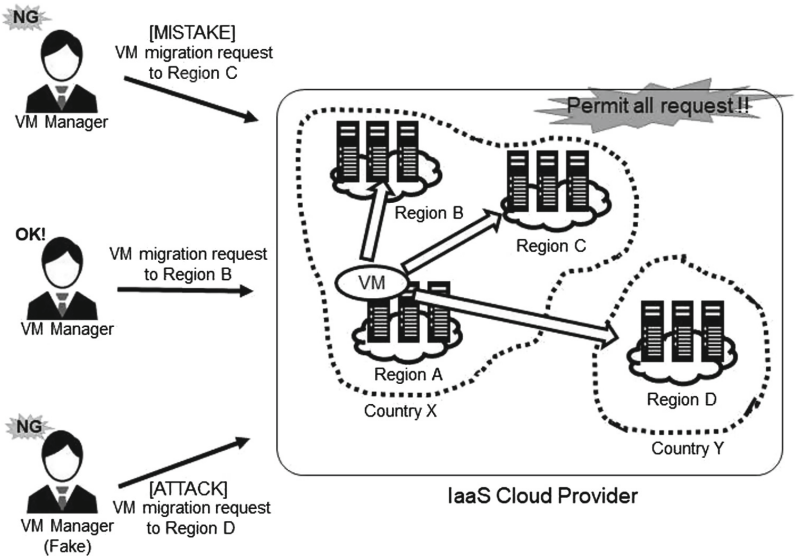

Fig. 1. VM migration problems.

smart contract refers to a program that operates on a blockchain network. Also, blockchain infrastructure can be divided into public type, consortium type and private type (Table 1). The public type, as represented by bitcoin, does not have a centralized management period, and an unspecified number of users can freely participate and participate in mining. Consortium type and private type are characterized by having a manager. Consortium types operate in multiple organizations, whereas private types operate in a single organization. Since mining can be controlled by the permission of the manager, it can be used for enterprise use such as financial system management.

Also, Blockchain was developed to support cryptocurrencies which can be used for all forms of trading without mediation. Transaction data is recorded as transaction content data and becomes a block. This data is open and anyone can check it. Also, the specific transaction content of the transaction is encrypted by the hash function. Hash functions have the irreversibility that the original data cannot be read from the generated string. Furthermore, the block data contains the transaction encrypted by the hash function and the hash value of the previous block data. This has the advantage that an attacker must compromise 51% of the system to exceed the hashing capability of the target network. Therefore, attacking blockchain networks is not computationally practical. In addition, there is a smart contract as one of the blockchain mechanism. Smart contracts, as the name implies, are protocols that can make contracts smart. It can automatically execute contract condition confirmation and fulfillment.

Table 1. Blockchain infrastructure types

	Public	Consortium (Multiple organizations)	Private (Single organization)
(Mining) Node type	No limit	Restrictable	Restrictable
Blockchain view	No limit	Restrictable	Restrictable
At block generation	High difficulty mechanism required	Any	Any
Mining reward	Necessary	Any	Any

4.2 Policy Control

We propose a policy based control mechanism to solve the problem of improper migration of data contained in VM. The control mechanism method is shown in Fig. 2. The VM manager describes the REGULATION that identifies the list of available countries and organizations based on the data regulations in the VM. The country and organization identifiers described in REGULATION are CountryCode and OrganizationCode. CountryCode indicates a country where data movement is permitted based on regulations. OrganizationCode represents an organization that instantiates and uses virtual machines. For example, if the host machine location is Japan and the organization is Company A, it described as "JP. Company A". If the data can be moved to all countries, CountryCode value is "All". If data is available to all organizations, OrganizationCode value is "All". If restrictions on the data in VM are changed, it calculates the difference between the list of country and organization identifiers added to the newly acquired data and the list of already described identifiers, and describe the values. Also, the host machine manager describes COUNTRY that identifies the country based on the physical area where the host machine is located. The country identifier described in the COUNTRY is CountryCode. If the host machine is located in Japan, it is described as "JP" Furthermore, the host machine manager describes ORGANIZATION as the organization that manages the host machine. The identifier of the organization described in ORGANIZATION is OrganizationCode. If Company B is the organization that manages the host machine, it described as "Company B".

When the migration is performed, compare the moving VM's REGULATION and the destination host machine's COUNTRY and REGULATION. First, check whether CountryCode of REGULATION is included in CountryCode of COUNTRY. Similarly, check whether the OrganizationCode of REGULATION is included in the OrganizationCode of ORGANIZATION. If both checks passed, migration is performed because the data acquired by the migrating VM can be migrated to the country or organization in which the migration destination host machine exists. The migration execution is the same as the conventional migration process. If it is not included, migration cannot be performed because it cannot move to the country where the destination host machine exists. By these mechanisms, we avoid the unintended data infringement caused by the migration

of the terms and conditions of the state, the regulations of the country, and the permission from the owner of various data with different owners.

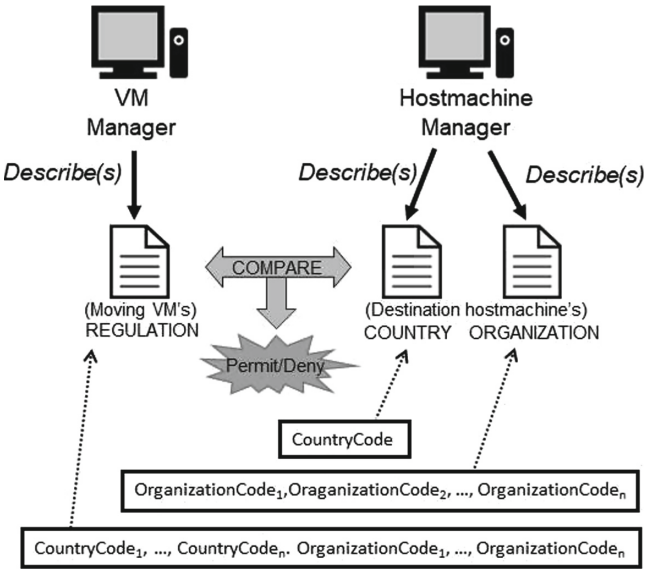

Fig. 2. Policy based control mechanism.

4.3 Control Mechanism Using Blockchain

The proposed mechanism could prevent the migration of inappropriate data, however if a malicious manager tampered with REGULATION, COUNTRY, or ORGANIZATION on a host machine running various VM, unintended migration is triggered and improper data movement occurs. Therefore, in order to prevent malicious falsification of data, we propose a method to prevent falsification using blockchain technology with a strong data protection mechanism.

The example environment using the proposed mechanism is shown the Fig. 3. Store REGULATION, COUNTRY, and ORGANIZATION in a blockchain. Only authorized VM managers and host machine managers can add or change these data. The VM manager registers REGULATION in the blockchain when the VM is newly deployed or the permission condition of the data in the VM is changed. The host machine manager registers COUNTRY and ORGANIZATION in the blockchain when a new host machine is installed or when the host machine affiliation is changed. In addition, the control mechanism using policy is implemented in the smart contract. The flow when the VM manager requests migration is shown below (Fig. 4).

1. VM manager requests migration to the blockchain.
2. Compare the moving VM's REGULATION and the destination host machine's COUNTRY and ORGANIZATION in the blockchain's smart contract.

3. If both checks pass, it requests the cloud to execute VM migration. If it does not pass the check, VM migration will not be performed.

The blockchain technology preserves the change history of REGULATION, COUNTRY, and ORGANIZATION, and checks if there is any tampering with past data. Since it is guaranteed that the current data has not been tampered based on the hash value of the past data, the person who is not the VM manager with the proper authority or the host machine manager cannot change the data. The blockchain distribution ledger is held by one host machine in each region, and the host machines in all regions hold the same information. For example, if a region exists from A to F, there will be 5 nodes holding the dispersion ledger. Only the VM manager and host machine manager can change the data in the distributed ledger.

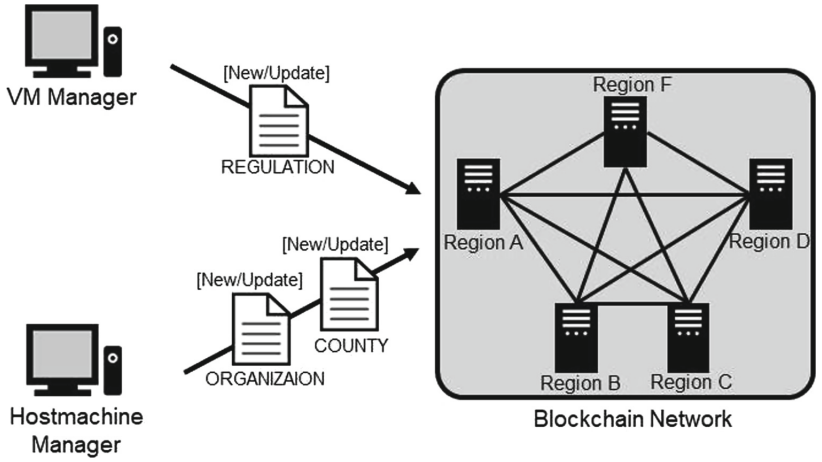

Fig. 3. Registers to blockchain.

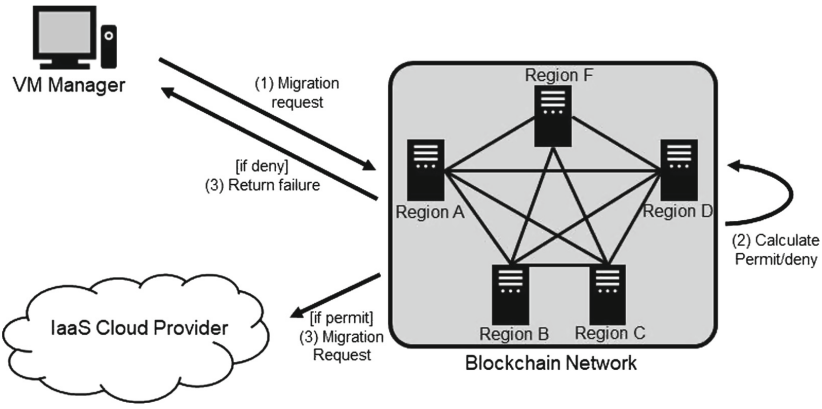

Fig. 4. Migration request process.

5 Implementation

5.1 OpenStack

On July 19, 2010, Rackspace and NASA announced the start of the project and released the initial version of OpenStack in October 2010. It features a completely open development style that follows the Ubuntu development style. It provides a multi-tenant type Infrastructure as a Service (IaaS) environment and can perform various operations on the IT infrastructure, such as creating virtual machines, creating networks, changing firewall policies, and so on. In addition, it supports cold migration and live migration, and can automatically move a VM running on a physical host machine to another physical host machine by typing a single command line or with a few mouse clicks on Horizon.

5.2 Hyperledger Fabric

In this paper, we focus on Hyperledger Fabric, which is a private type blockchain. Hyperledger Fabric is a foundation of blockchain developed by open source community about blockchain technology promoted by the Linux Foundation. In particular, it focuses on enterprise level usage, and each component can be customized freely. Figure 5 shows the configuration of Hyperledger Fabric. A control mechanism using the proposed policy is implemented in chaincode (smart contract), and a secure control mechanism is realized by storing data in PeerLedger in KVS format.

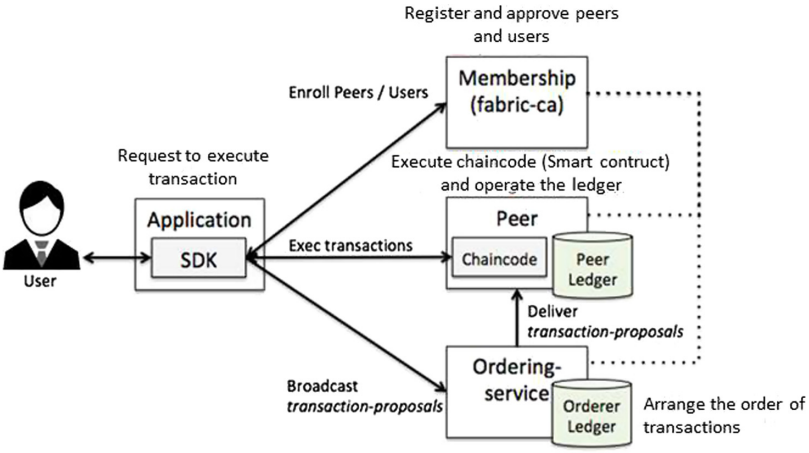

Fig. 5. Hyperledger Fabric configuration.

5.3 Constitution

In this section, we implemented the control mechanism using blockchain which we proposed in Sect. 4. This mechanism can be applied to various types of cloud environments without changing the migration process.

The configuration of the implementation environment is shown in Fig. 6. The data protection mechanism consists of "VM Manager", "FrontEnd" and "Cloud Provider". "Cloud Provider" consists of 5 host machines with 3 regions. BCMH (Blockchain Management Host) is installed on HostMachine01, HostMachine03, and HostMachine04, and has two roles for cloud's host machine and blockchain node. Also, VM1 is running on HostMachine02. "FrontEnd" performs operations on blockchain environment and operations on the cloud environment. The migration execution request is not made by the "VM Manager" directly to "Cloud Provider" but by the migration request to "FrontEnd". "FrontEnd" consists of a blockchain client and a cloud management client. Blockchain clients form a blockchain network with BCMH in "Cloud Provider". The cloud management client can operate the SDK of "Cloud Provider" and creates a migration request to "Cloud Provider". Specifically, it is used to manipulate the OpenStack API.

Also, the configuration of each host machine and VM1 is shown in Table 2. "Cloud Provider" uses the one built with OpenStack Tika. The blockchain uses Hyperledger Fabric 1.2 and Hyperledger Composer 0.20.1. The host machine and VM1's OS use CentOS7. The host machine consists of 4 core CPU, 12 GB memory, and 100 GB storage. VM1 consists of 1 core virtual CPU, 2 GB virtual memory, and 20 GB virtual storage. Also, it is assumed that Apache 2.0 and MySQL are installed in VM1, and simple web service is running. Table 3 shows the attributes of each host machine. Assign 3 regions, 3 CountryCodes, and 2 OrganizationCodes to each host machine.

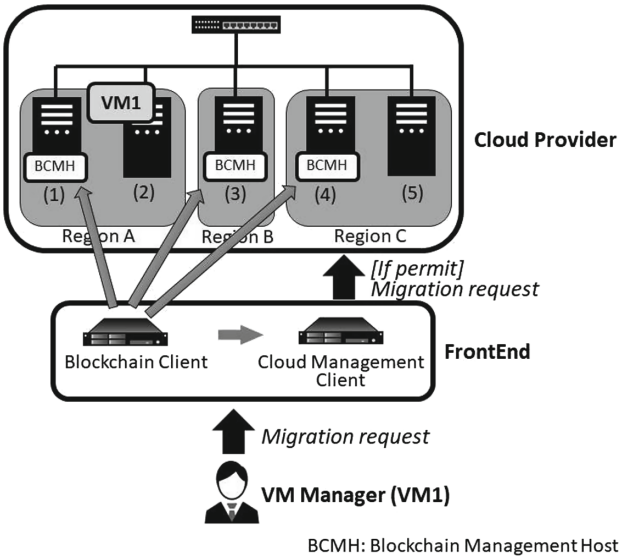

Fig. 6. Implementation environment configuration.

Table 2. Hostmachine and VM1 configurations.

Name	OS	CPU	Memory	Storage	IP address
Host Machine(1)	CentOS7	4 core	12 GB	100 GB	10.0.0.161
Host Machine(2)	CentOS7	4 core	12 GB	100 GB	10.0.0.162
Host Machine(3)	CentOS7	4 core	12 GB	100 GB	10.0.0.163
Host Machine(4)	CentOS7	4 core	12 GB	100 GB	10.0.0.164
Host Machine(5)	CentOS7	4 core	12 GB	100 GB	10.0.0.165
FrontEnd	CentOS7	4 core	12 GB	100 GB	10.0.0.170
VM1	CentOS7	1 core	2 GB	20 GB	10.0.0.7

Table 3. Hostmachine attributes.

Host name	Region	CountryCode	OrganizationCode
Host Machine(1)	A	JP	CompanyA CompanyB
Host Machine(2)	A	JP	CampanyA
Host Machine(3)	B	DE	CompanyA CompanyB
Host Machine(4)	C	GB	CompanyC
Host Machine(5)	C	GB	CampanyB

6 Evaluation

We evaluate the overhead cost of the mechanism in migration using an environment that implemented control mechanisms using policy in Sect. 5. In addition, since cold migration has fewer time constraints, this evaluation covers live migration.

6.1 New/Update Register Request

This measures execution time when new/update REGULATION, COUNTRY, or ORGANIZATION to a blockchain is performed. The Listing 1.1 below shows the JSON when a VM1 manager register a policy to a blockchain. The policy type is REGULATION, the name of the VM targeted by the policy is VM1, the country in which the VM permits are JP and GB, and the organization in which the VM is permitted CompanyA and CompanyB. Listing 1.2 and Listing 1.3 below shows the JSON when the hostmachine(1) manager registers the policy to the blockchain. In the case of COUNTRY, the type of policy is COUNTRY, the name of the hostmachine is hostmachine01, and the country in which the hostmachine is located is JP. In the case of ORGANIZATION, the type of policy is ORGANIZATION, the hostmachine name is hostmachine(1), and the hostmachine organization is CompanyA and CompanyB.

The time taken for registration to the blockchain was measured. Information on different values was registered twenty times for each policy, and the average

time was examined. It took 2210 ms on average from when the VM manager or hostmachine manager made a request for policy registration to the blockchain and before processing completion notification was returned. Since these policies are not updated frequently, it takes more than 2 sec, but there is no problem.

Listing 1.1. Register REGULATION policy in VM1

```
{
  ''type'': ''REGULATION'',
  ''vmName'': ''vm1'',
  ''country'': ''JP,GB'',
  ''organization'': ''CompanyA,CompanyB''
}
```

Listing 1.2. Register COUNTRY policy in Hostmachine(1)

```
{
  ''type'': ''COUNTRY'',
  ''hmName'': ''hostmachine(1)'',
  ''country'': ''JP''
}
```

Listing 1.3. Register ORGANIZATION policy in Hostmachine(1)

```
{
  ''type'': ''ORGANIZATION'',
  ''hmName'': ''hostmachine(1)'',
  ''organization'': ''CompanyA,CompanyB''
}
```

6.2 Live Migration Request

Here, we evaluate the overhead cost of the mechanism in VM migration. In addition, live migration is targeted in this evaluation. The behavior when the VM1 manager request a live migration to a different region's hostmachine, and the time required for processing the blockchain when live migration is permitted and the time required for live migration is measured.

First, we evaluate live migration (Table 4). The manager of VM1 running on the hostmachine(1) request live migration to hostmachine in other regions. When live migration is requested to the hostmachine(3), deny is returned and live migration is not performed. This is because the country where VM1 have permits are JP and GB, whereas the hostmachine(3) is located in DE, which is not permitted. Next, when the live migration request to the hostmachine(4), deny is returned and live migration is not performed. This is because the organization that VM1 have permits are CompanyA and CompanyB, and none of them is included in the organization of the hostmachine(4). Next, when live migration is requested to the hostmachine(5), permit is returned and live migration is

executed. This is because all the countries and organizations that VM1 permits are included in the hostmachine(5).

Next, measure the time when the manager of VM1 that is running on hostmachine(1) requests live migration to hostmachine(5) shown Fig. 7. As mentioned above, live migration is permitted by the proposed mechanism in blockchain. The request for live migration was made 20 times and the average time was examined under the same conditions. After VM1 manager requested live migration to the blockchain, it took 15633 ms on average before processing completion notification is returned. The breakdown was 91 ms for blockchain processing and 15542 ms for live migration processing. Compared to the processing time of live migration, the processing time in the blockchain is 0.58%, and there is almost no overhead cost.

Table 4. Evaluate live migration request of VM1.

	Hostmachine(1)	Hostmachine(2)	Hostmachine(3)	Hostmachine(4)	Hostmachine(5)
Hostmachine(1)	Permit	Permit	Deny	Deny	Permit
Hostmachine(2)	Permit	Permit	Deny	Deny	Permit
Hostmachine(3)	Deny	Deny	Deny	Deny	Deny
Hostmachine(4)	Deny	Deny	Deny	Deny	Deny
Hostmachine(5)	Permit	Permit	Deny	Deny	Permit

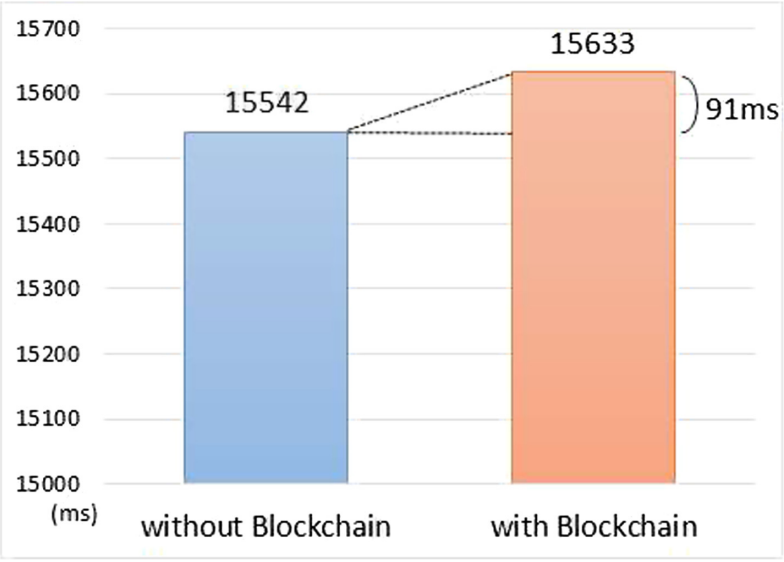

Fig. 7. Measure the VM live migration time when hostmachine(1) to hostmachine(5)

7 Conclusion

We proposed a mechanism to ensure compliance with the conditions granted by the data owner, the country's laws and regulations, and the regulations of the organization during VM migration. By constructing the proposed mechanism in a blockchain, it is possible to prevent unintended data movement and malicious tampering and enable robust VM migration control. We implemented the proposed mechanism to the existing cloud environment together with the blockchain. Policies on VM and hostmachines were registered in the blockchain, and it was evaluated whether the decision of VM live migration based on the policy was correctly performed. We also measured the execution time of live migration using the proposed mechanism and showed that there was almost no overhead cost.

Acknowledgements.. This work was supported by JSPS KAKENHI Grant Number JP20K19778.

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Tan, L., Wang, N.: Future internet: the Internet of Things. In: 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. V5-376–V5-380 (2010)
3. Dhananjay, S., Tripathi, G., Jara, A.J.: A survey of Internet-of-Things: future vision architecture challenges and services. In: IEEE World Forum on Internet of Things (WF-IoT), pp. 287–292 (2014)
4. General Data Protection Regulation: GDPR. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed Jun 2019
5. AWS. https://aws.amazon.com/. Accessed Jun 2019
6. Microsoft Azure Cloud Computing Platform & Service. https://azure.microsoft.com/. Accessed Jun 2019
7. Khajeh-Hosseini, A., Greenwood, D., Sommerville, I.: Cloud migration: a case study of migrating an enterprise IT system to IaaS. In: IEEE 3rd International Conference on Cloud Computing, pp. 450–457 (2010)
8. Shetty, J., Anala, M.R., Shobha, G.: A survey on techniques of secure live migration of virtual machine. Int. J. Comput. Appl. (IJCA) **39**(12), 34–39 (2012)
9. Aiash, M., Mapp, G., Gemikonakli, O.: Secure live virtual machines migration: issues and solutions. In: Advanced Information Networking and Applications Workshops (WAINA), pp. 160–165 (2014)
10. Upadhyay, A., Lakkadwala, P.: Secure live migration of VM's in cloud computing: a survey. In: Reliability, Infocom Technologies and Optimization (ICRITO), pp. 1–4 (2014)
11. Rathod, N., Chauhan, S.: Survey: secure live VM migration in public cloud. Int. J. Sci. Res. Dev. (IJSRD) **2**(12), 271–274 (2015)
12. Gutierrez-Garcia, J.O., Ramirez-Nafarrate, A.: Policy-based agents for virtual machine migration in cloud data centers. In: Services Computing (SCC), pp. 603–610 (2013)

13. Koto, A., Kono, K., Yamada, H.: A guideline for selecting live migration policies and implementations in clouds. In: Cloud Computing Technology and Science (CloudCom), pp. 226–233 (2014)
14. Cui, L., Tso, F.P., Pezaros, D.P., Jia, W.: PLAN: a policy-aware VM management scheme for cloud data centres. In: Utility and Cloud Computing (UCC), pp. 142–151 (2015)
15. Papadopoulos, A.V., Maggio, M.: Virtual machine migration in cloud infrastructures: problem formalization and policies proposal. In: Decision and Control (CDC), pp. 6698–6705 (2015)
16. Xu, X., et al.: A taxonomy of blockchain-based systems for architecture design. In: IEEE International Conference on Software Architecture (ICSA 2017), pp. 243–252 (2017)
17. Dorri, A., Kanhere, S., Jurdak, R., Gauravaram, P.: Blockchain for IoT security and privacy: the case study of a smart home. In: The 2nd IEEE Workshop on Security, Privacy, and Trust in the Internet of Things (PERCOM), pp. 618–623 (2017)
18. Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., Sassone, V.: Blockchain-based database to ensure data integrity in cloud computing environments. In: Italian Conference on Cyber security (ITASEC 2017), pp. 146–155 (2017)
19. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L.: ProvChain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: The 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 468–477 (2017)
20. Xu, C., Wang, K., Guo, M.: Intelligent resource management in blockchain-based cloud datacenters. IEEE Cloud Comput. **4**(6), 50–59 (2017)
21. Sharma, P.K., Chen, M., Park, J.H.: A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access **6**, 115–124 (2018)
22. Rahalkar, C., Gujar, D.: Content addressed P2P file system for the web with blockchain-based meta-data integrity. In: 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), pp. 1–4 (2019)
23. KVM. https://www.linux-kvm.org. Accessed Jun 2019
24. Xen Project. https://xenproject.org/. Accessed Jun 2019
25. Compute Engine - IaaS—Compute Engine. https://cloud.google.com/compute/. Accessed Jun 2019
26. Build the future of Open Infrastructure. https://www.openstack.org/. Accessed Jun 2019