# Evaluation of Highly Available Safety Confirmation System Using an Access Prediction Model

Masaki Nagata[†1†2*], Yusuke Abe[†2], Misato Fukui[†2], Chihiro Isobe[†2], and Hiroshi Mineno[†1**]

[†1]Graduate School of Science and Technology, Shizuoka University, Japan
[†2]AvanceSystem Corporation, Japan
[*] nagata@avancesys.co.jp
[**] mineno@inf.shizuoka.ac.jp

***Abstract*** - A safety confirmation system provides a mechanism for sharing users' safety information in disasters, and is therefore required to operate reliably in the event of a disaster. Further, it is essential that the architecture is able to expand to accommodate additional resources during disasters because it is accessed by many users at such times. Increasing the appropriate resources during disasters necessitates the use of access prediction based on the access distribution during past disasters. Many conventional safety confirmation systems utilize a Relational Database (RDB) because the RDB structure is suitable for data management. However, because RDB has weak partition-tolerance characteristics it has availability issues. In this paper, we propose a method that improves the partition-tolerance using multiple servers, and an access prediction method that utilizes lognormal distribution to predict access to safety confirmation systems during disasters. The proposed method also employs a distributed database system with multiple servers and access prediction is carried out using a plurality lognormal distribution that depends on the time at which a disaster occurs. The results of evaluations conducted indicate that the proposed method improves availability and allocates the appropriate resources for access distribution during disasters.

***Keywords***: access prediction, lognormal distribution, distributed database system, load balancing, safety confirmation system

## 1 INTRODUCTION

The ability to share safety information with users during disasters that result in serious damage and life-threatening danger, such as the Great East Japan Earthquake of 2011 and the 2016 Kumamoto Earthquake, is important because the early collection and disclosure of user safety information can save many lives. A safety confirmation system provides a means of sharing information with users during disasters [1]. A safety confirmation system is a web system that collects and presents safety information during disasters from and to users registered in the system. For example, the disaster bulletin board of a telecommunications carrier, Google Person Finder [2], and J-anpi [3] can crossover and collectively search the safety information they each have available. These systems are suitable for implementation using a web system, because a web system is accessible by PC and smartphone for reporting and presenting safety information. In addition, the system operation infrastructure can be outsourced to a cloud vendor that has disaster countermeasures rather than a single company's on-premises assets because a safety confirmation system is required to operate continuously during a disaster. However, migrating a system to the cloud environment is problematic.

The first issue is that of distributed data management for system redundancy. Because a safety confirmation system is required to operate continuously and reliably during disasters, its data management has to include strong partition-tolerance that enables alternate operation on another server when the primary server is down. Fu [4] proposed a method that improves availability using a redundant server to configure the system. In addition, we previously proposed a general safety confirmation system with global redundancy; that is, with servers in multiple regions, overseas as well as domestic. The use of multiple servers enables inevitable operation as a distributed system. Moreover, the study of conventional safety system [5] [6] that contains the author's previous studies is using an RDB for data management. However, that conventional safety confirmation system uses an RDB for data management, which poses a problem as an RDB has weak partition tolerance.

The second issue is that of adjusting the number of servers in accordance with the access situation. Because a safety confirmation system has very high access traffic during disasters and very low traffic when there is no disaster, the number of servers utilized should vary accordingly in order to reduce the operating cost. Access to the safety confirmation system increases during disasters; hence, the ability to determine the number of servers suitable to accommodate access traffic during a disaster is important. We obtained an understanding of the tendency of access traffic during disasters by analyzing access distribution during past disasters. As a result, real access traffic was found to exhibit a lognormal distribution. Consequently, we previously proposed an access prediction model that uses lognormal distribution [7]. The access prediction model showed that the cost of using additional servers can be reduced by allocating an appropriate number of servers for access distribution that varies with time during a disaster. However, access prediction is problematic in that it depends on the disaster situation. Thus, to overcome these issues, in this paper we propose a method that uses a distributed database with multiple servers and access prediction using a plurality lognormal distribution. We demonstrate the effectiveness of prototype safety confirmation system with these functions implemented.

## 2 RELATED WORK AND ISSUES

### 2.1 Safety Confirmation System

Work related to safety confirmation systems has been reported in various fields, e.g., information collection and sharing, network communication, and web systems. As regards information collection and sharing, Ishida et al. [8] proposed a safety information system that gathers and shares refugee information between different evacuation centers set up by each local government during a disaster. Registration of refugee information is accomplished using a personal IC card issued to each user and a reader. This is in consideration of children and older people inexperienced with ICT equipment. As regards network communication, Wang et al. [9] proposed a system that uses the AODV protocol to enable communication between users using smartphones. The system enables reliable transmission of safety information using node-to-node communication when the communication infrastructure is damaged or usage of communication resources is restricted.

The subject of this study is a general safety confirmation web system. The process followed by a safety confirmation system is as follows (Fig.1). First, the meteorological information service provides information about the occurring disaster to the safety confirmation system. Then, during the disaster, the safety confirmation system sends an e-mail to prompt users for safety confirmation. Next, users who receive the e-mail report their safety information to the safety confirmation system. Finally, users share their safety information with each other. Yuze and Suzuki [10] proposed relocating safety confirmation systems running on on-premises equipment to the cloud environment to improve service availability and to ensure sustained operation should the on-premises environment be adversely affected during the disaster. Echigo et al. [11] proposed load balancing and redundancy by mirroring using multiple servers to improve robustness. Thus, web systems have generally been used for information management in communication and information gathering related work. Therefore, it is clear that sustainable operation of the web system infrastructure is important to achieve effective overall safety information management during disasters.

### 2.2 Issues: Distributed Data Management

In a conventional safety confirmation system, data are managed using an RDB because the Create, Read, Update, Delete (CRUD) operation of each attribute data with a key such as user ID is suitable for managing users' safety information and department data. However, access traffic to each piece of attribute data is usually low; much of the access is the safety report during the disaster. Safety report access is an Update operation to update users' safety information data. The RDB sharding technique for dispersed access using multiple servers is an advantage but RDB has problems such as data search complexity and change of the ID numbering of the hash function associated with the data scale. In addition, RDB is not suitable for distributed systems because it

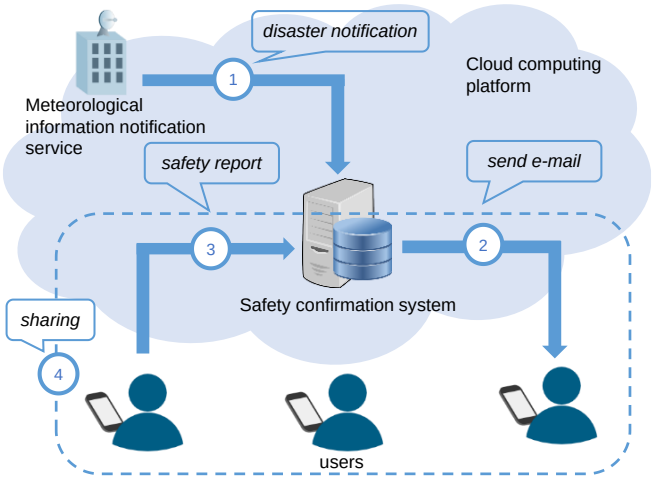

Figure 1: Flow of the safety confirmation system

is vulnerable to partition-tolerance of the CAP theorem. By contrast, a safety confirmation system should use distributed data management that runs on another server when the main server is down because continuous operation is essential. Therefore, it is necessary to improve availability using multiple servers and distributed data management that is able to manage high-volume access traffic during disasters.

### 2.3 Issues: Number of Servers in Accordance with the Situation

The cost of safety confirmation systems, which differs depending on the number of users accessing the system normally and during disasters, can be reduced by operating the number of servers in accordance with the access situation. The most simplistic resource management is to continue running the system on a large number of servers, regardless of the situation. However, the smaller amount of access traffic during when there is no disaster means that the continuous operation of many servers at all times results in surplus resources, and, consequently, surplus costs. Therefore, if the required number of servers can be ensured to be in accordance with the access situation, this would be ideal for the resource management of the safety confirmation system. It would reduce the cost when there is no disaster, when the amount of access traffic is small. Moreover, calculating and allocating the appropriate number of servers before access concentration is desirable to avoid impairing user convenience when the response performance decreases. Calculation of a suitable number of servers in accordance with the access situation necessitates prediction of the access distribution to the system during disasters. In our previous study, we proposed an access prediction model that uses a lognormal distribution to predict access to the safety confirmation system during disasters. However, this approach is problematic as the use of a single lognormal distribution to model access prediction is difficult. This is because the access distribution trend to the system was found to differ according to the time at which a disaster occurs. Therefore, it is necessary to calculate the number of servers by selecting the appropriate access prediction model in accordance with the disaster occurrence time.

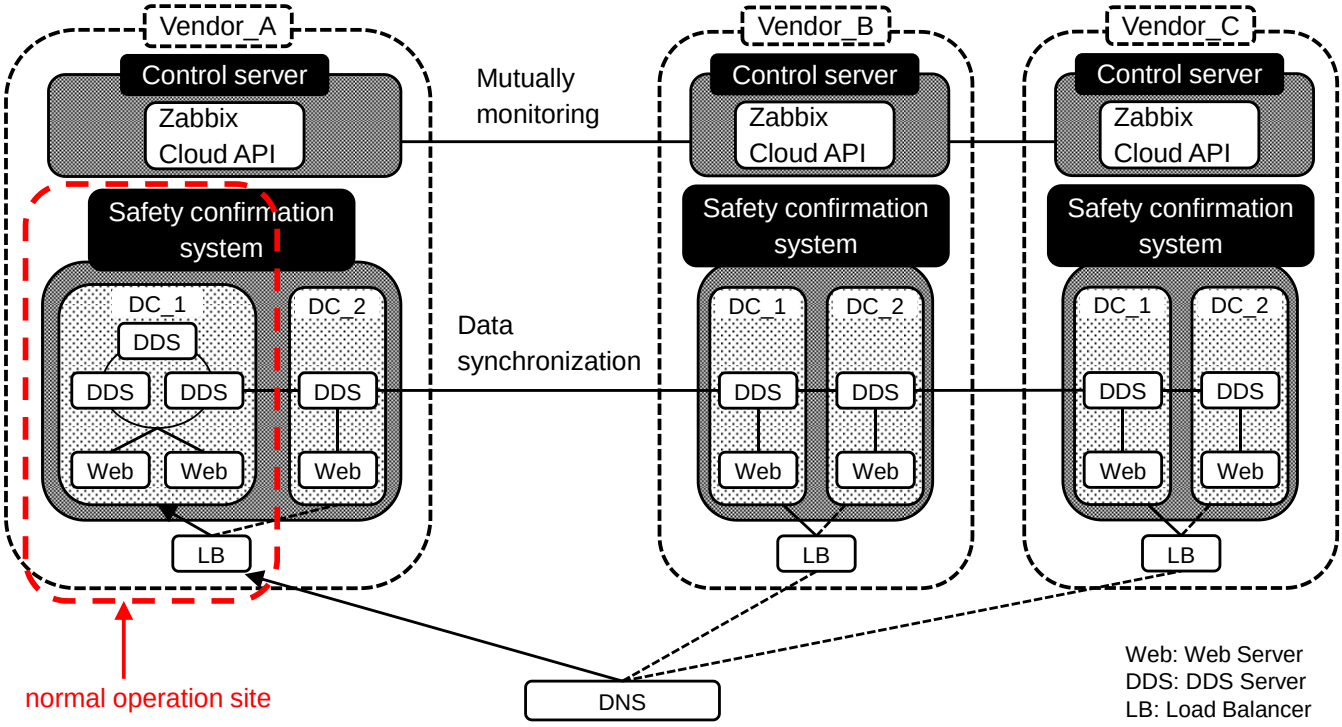


Figure 2: Architecture of the proposed system

## 3 PROPOSED SYSTEM

### 3.1 System Overview

The proposed system operates in an intercloud environment using three cloud vendors (Fig.2): vendor A, Amazon Web Services (AWS) [12]; vendor B, Microsoft Azure [13]; and vendor C, Cloudn [14]. DC_1 and DC_2 are the service provision regions of the vendors, each of which has multiple data centers. The system availability was improved by the monitoring from each vendor. During normal operation of the system, all accesses are directed to vendor A and vendor A is in charge of load balancing against increased access during a disaster. Vendor B and C are backup sites for vendor A. When a failure occurs, failover is accomplished by changing the access destination to vendor B (or C). The safety confirmation system and control server is deployed to each vendor. The safety confirmation system consists of a web server and a Distributed Data System (DDS). The DDS is a mechanism for distributed management of the data in cooperation with multiple servers. Distributed data management is implemented by arranging a plurality DDS node to each vendor. Data synchronization is carried out by using a data replication function. The control server uses the access prediction model to calculate the appropriate number of servers required during the disaster, and to scale out the web server for the safety confirmation system in order to conduct load balancing. In addition, it monitors each vendor and uses Zabbix [15] to perform failover when failures occur. Incidentally, the data synchronization of each vendor and the recovery flow in the event of failure are not discussed in this paper as those have already been expounded on in [7].

An overview of the safety confirmation system is shown in Fig.3. The figure depicts operation by multiple customers to share the server resources and usage by registered users. A summary of the operation after the occurrence of the disaster that corresponds to the region and the earthquake intensity threshold set by the customer unit is sent by e-mail to promote the safety report to the target users. Figure 3, for example, shows that an earthquake of intensity five upper occurred in Tokyo and Kanagawa, and that the number of target users is 15,700 among customers A, B, and D. Specifically, the number of target users of the proposed system changes according to the scale of the disaster. The system performs access prediction and load balancing in accordance with the number of target users.

Figure 4 shows the load balancing flow using the access prediction model. Inserting an additional server is called a scale-out operation, whereas reduction is a scale-in operation. The scale-out operation is not executed if the server is acceptable with the normal configuration of servers for the number of target users at the disaster; if unacceptable, scale-out executes with the appropriate number of servers based on the access prediction model. Load balancing is executed by scaling-out the web server in units of two servers, one for each of the two locations in "Vendor A: DC_1: AWS." Thus, it will add two, four, six–an even number of web servers. The proposed system equally distributes the load by utilizing the same number of web servers in each data center via the load balancer. Moreover, an e-mail is sent to target users following scale-out completion to avoid access concentration before the construction of a load balancing environment.

In order to execute a scale-out, it is necessary to ascertain the load point of the system. This is because it is possible to improve the processing power by adding a server when it accepts a certain load in terms of system resources.

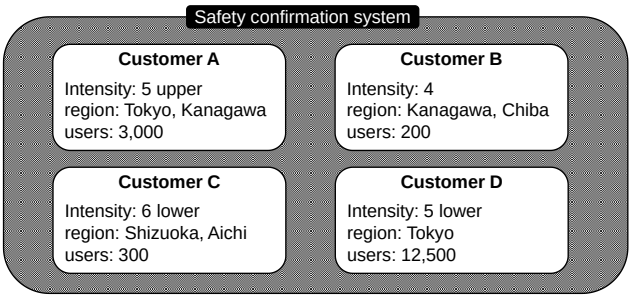


Figure 3: Multiple customer operation

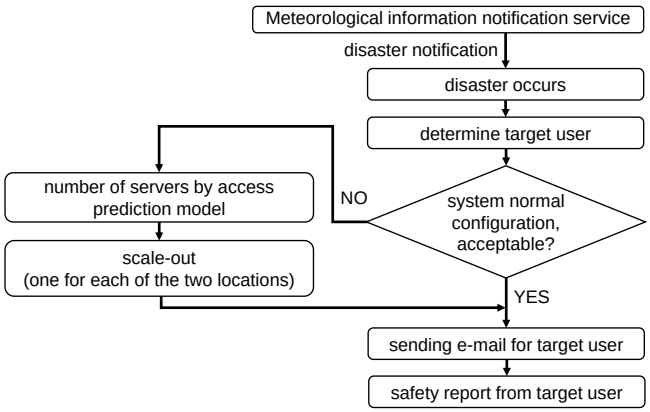


Figure 4: Flow of load balancing

Therefore, it is necessary to ascertain the load point of the system resources of the safety confirmation system during a disaster. Access to the system during a disaster accounts for more than 90% of the safety report accesses, based on access logs. Thus, the load point of the system is the safety report access concentration at the disaster. We measured the resource consumption of the load point using JMeter [16] to create a test scenario for safety report access. JMeter is a set of evaluation tools that enables a target system to be accessed via the web. Figure 5 shows the results of measuring each of the resources by changing access to the safety report every 10 minutes. The web server used AWS EC2 [17] t2.small and the DB server used EC2 c3.xlarge. Figure 5 shows that the web server CPU usage increased significantly with increasing safety report access traffic and each of the resource loads is considerably less than the web server CPU usage. This result indicates that the load point of the safety confirmation system increases web server CPU usage because of the safety report access concentration at the disaster. Therefore, the proposed system performs scale-out and scale-in by monitoring web server CPU usage. It should be noted that, strictly speaking, the database server was assigned a load, but this paper only targets the web server to simplify the explanation.

Server types and number of servers to be used in scale-out and scale-in are decided based on single server processing power. Murta and Dutra [18] modeled the resource management of an entire system from the benchmark result of a single server. In this paper, we calculate the appropriate number of servers based on the processing of a single server to determine the access traffic obtained in the prediction. The access processing power of this study is determined using the safety report access that can be processed per unit

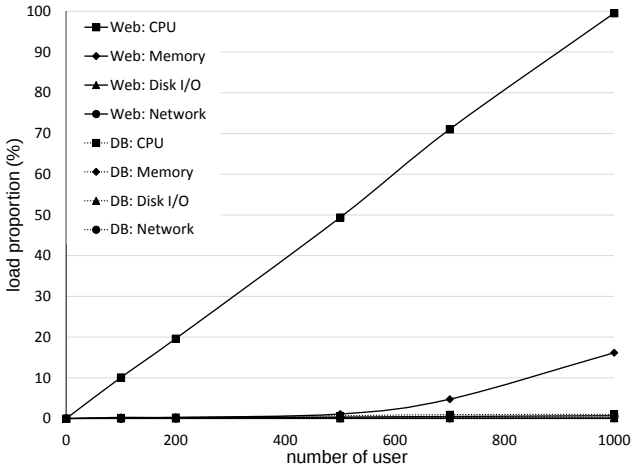


Figure 5: Web, DB resource usage proportion

Table 1: EC2 instance types: UnixBench results

| Instance Type | vCPU | Memory (GiB) | System Benchmarks Index Score | Costs ($,hour) |
|---|---|---|---|---|
| t2.small | 1 | 2 | 1702.30 | 0.034 |
| t2.medium | 2 | 4 | 2536.00 | 0.068 |
| t2.large | 2 | 8 | 2537.20 | 0.136 |
| m3.medium | 1 | 3.75 | 848.90 | 0.077 |
| m3.large | 2 | 7.5 | 1858.60 | 0.154 |
| m3.xlarge | 4 | 15 | 2945.00 | 0.308 |
| m4.large | 2 | 8 | 2025.00 | 0.14 |
| m4.xlarge | 4 | 16 | 3132.80 | 0.279 |

time. Each vendor has a variety of server types; measurements were conducted with respect to AWS, vendor A, which is the main one that performs load balancing. This was measured to clarify the relationship between the access processing power and each EC2 instance type. The measurement method is the same for each vendor. Table 1 shows the UnixBench [19] measurement results for a general-purpose EC2 instance type. The overall performance index of UnixBench is provided by the "System Benchmarks Index Score." Table 1 shows that the "System Benchmarks Index Score" increased with EC2 "vCPU." However, the "System Benchmarks Index Score" is not simply doubled when "vCPU" is doubled. Thus, the cost performance is higher for one vCPU than for two vCPUs. Therefore, the proposed system adopted t2.small from among the available "vCPUs," considering the cost per hour and result of the "System Benchmarks Index Score." Incidentally, AWS EC2 defines the baseline of CPU usage for the t2 series, including t2.small. If the CPU usage is above the baseline, the state becomes burst. Burst is a state in which CPU performance is temporarily reduced; it is able to continue by consuming the AWS CPU credits. If the CPU credits are exhausted, CPU performance cannot exceed the baseline. In this study, the processing power of one server was determined by the number of safety report accesses at a CPU usage of 20%, the baseline for t2.small, not considering the processing power of the burst. CPU usage at 20% of t2.small is able to process 200 safety report accesses in 10 minutes, as shown in Fig.5. Moreover, the processing power

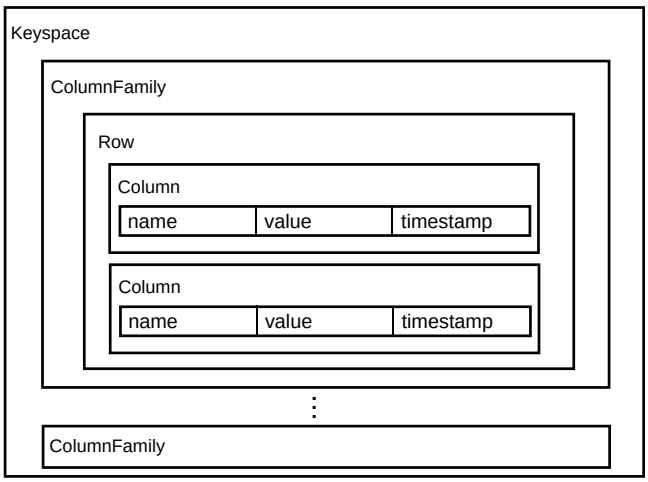


Figure 6: Data structure of cassandra

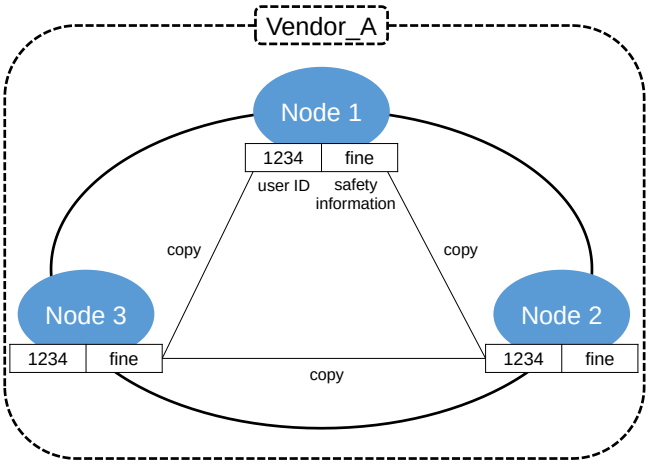


Figure 7: Node and replication factor

of the system with normal configuration is 400 safety report accesses in 10 minutes, because there are two t2.small instances for each web server.

## 3.2 Distributed Data Management Using Cassandra

Data management of the proposed system that converts the RDB data schema is desirable, because conventional safety confirmation systems use RDB for data management. Therefore, for data management of the proposed system Cassandra [20], which requires a schema definition among the distributed data management systems, was chosen. Cassandra is a NOSQL distributed data management system that also has excellent writing characteristics [21]. The data management structure of Cassandra is the Key-Value (KVS) method for managing a unique label (Key) for the data (Value). Cassandra is also a column-oriented NOSQL system. The column-orientation is obtained in an advanced simple KVS manner. It allows for multiple management of a set of Key and Value, which is referred to as a Column in Row, whereas simple KVS is managed in a one-to-one relationship between the Key and Value. The data structure of Cassandra is shown in Fig.6. The data units of Cassandra and RDB correspond in the following manner. Keyspace is database,

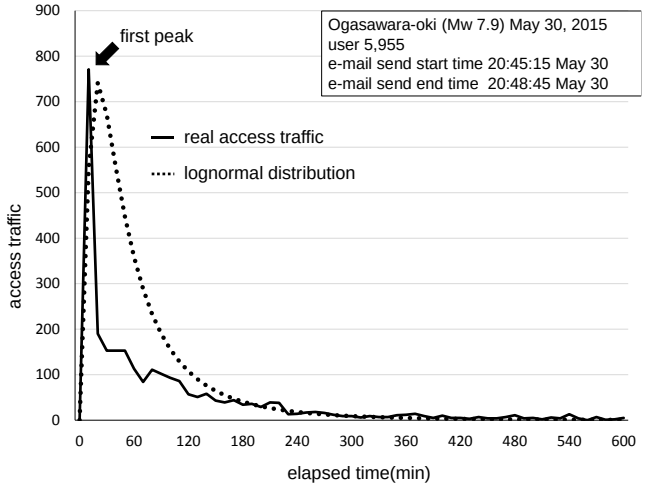


Figure 8: One-peak access traffic and lognormal distribution

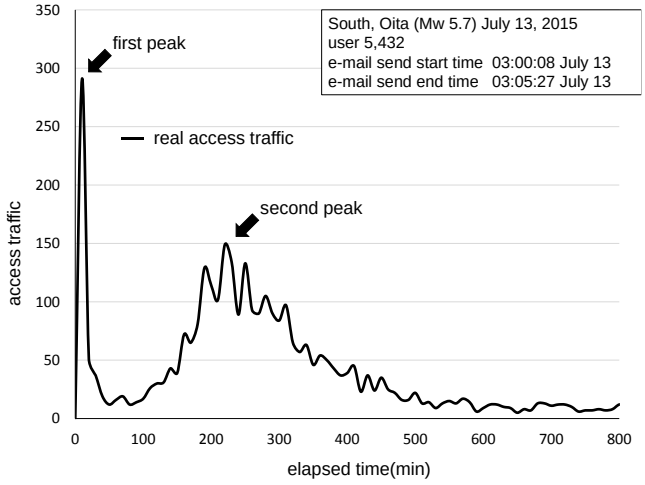


Figure 9: Two-peak access traffic

ColumnFamily is table, and Row is record. Porting of the data management is more easily done than other NOSQL systems because this structure is similar to the user management schema in RDB in the safety confirmation system.

Cassandra is typically operated on a cluster using multiple servers, rather than a single server. Operation in a cluster configuration enables continuous operation and improves availability as another node is alternatively operated when a node goes down. As shown in Fig.7, the proposed system operates in three nodes on vendor A, with Replication Factor (RF) = 3. Nodes from vendors B and C are used as backup. The RF is the total number of copies of the data. As shown Fig.7, in the case where RF = 3, the data have the user ID in Key and the safety information in Value to keep a copy of the data in three nodes. The proposed system improves the availability by copying the data to all nodes because the number of nodes is three RF = 3. It prepares for disaster recovery using the multi-data center capabilities of Cassandra so that nodes of vendor A and nodes of vendor B and C are involved in the automatic replication. Cassandra improves the availability by setting the number of nodes and RF properly. Therefore, it is suitable for data management of systems that require continuous operation, such as the safety confirmation system.

## 3.3 Access Prediction Model

### 3.3.1 Characteristic Access Distribution of the Safety Confirmation System

The prediction of access requires an understanding of the characteristics of access distribution with respect to the safety confirmation system. Figure 8 shows the access distribution during a disaster. The access traffic during the disaster reaches a peak a short while after the initial e-mail is sent by the safety confirmation system, and then decreases with time. Moreover, Fig.8 shows the lognormal distribution and access traffic during the disaster. To model the counting data, Poisson distribution is usually utilized. However, we propose using a lognormal distribution to predict access to the safety confirmation system during disasters, because we previously confirmed a certain normality via a normality test of access distribution in a disaster in our previous study [7]. When using this method, access is in accordance with a lognormal distribution with decay period from peak, and we achieved the expected effect in calculating the appropriate number of servers for access prediction; however, depending on the disaster situation, a single lognormal distribution is problematic.

Figure 9 shows the access traffic of a disaster that occurred at 3:00 at night. Figure 9 shows two peaks, with the first peak being immediately after the occurrence, and the second peak a few hours after the occurrence. The second peak is reached in the morning and reflects human activity time. This denotes that users who were sleeping during the disaster only accessed the system after awakening. Therefore, our proposed access prediction method uses a plurality lognormal distribution for access distribution consisting of two peaks.

### 3.3.2 Suitability with the Mixed Lognormal Distribution for Two Peaks

Access prediction is carried out by modeling the access trend distribution analysis using data from past disasters. Our previous study entailed access prediction with a single lognormal distribution model for a one-peak disaster, as shown in Fig.8. Lognormal distribution is defined as in Eq. (1), mode $M$ is Eq. (2), expected value $E$ is Eq. (3), where $\mu$ is the expected value of the normal distribution, and $\sigma$ is the standard deviation of the normal distribution. Each parameter of the lognormal distribution is determined by analyzing past disasters and disaster drill data. As detailed information can be found in [7], only an outline is given here. Mode $M$ is the time with the largest number of accesses and has fixed value of *20*. Expected value $E$ is the time of the average number of accesses and is calculated from the relation between the number of target users *TU* and mode $M$. Further, solving the simultaneous equations of Eq. (2) and Eq. (3) results in Eq. (4) and Eq. (5). Then, substituting $M$ and $E$ into Eqs. (4) and (5) gives $\mu$ and $\sigma$, the parameters in Eq. (1). Equation (6) is the access prediction model, which is used to predict the access number $AN$ at a time of $x$ minutes. $A$ is a coefficient of Eq. (1) for matching the peak access according to the number of target users *TU*.

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \quad (1)$$

$$M = \exp(\mu - \sigma^2) \quad (2)$$

$$E = \exp(\mu + \frac{\sigma^2}{2}) \quad (3)$$

$$\mu = \frac{(\ln(M) + 2 * \ln(E))}{3} \quad (4)$$

$$\sigma^2 = \frac{2 * (\ln(E) - \ln(M))}{3} \quad (5)$$

$$AN = A * f(x) \quad (6)$$

Toriumi et al. [22] conducted an analysis using the mixed lognormal distribution model for the concentration of multiple tweets from Twitter [23] during a disaster. This study examined the suitability of applying the mixed lognormal distribution for access distribution with two peaks with reference to previous research. The mixed lognormal distribution represented in Eq. (7) is based on Eq. (1). Moreover, it has two lognormal distributions considering its adaptation of two peaks. *f(x)* is the first peak lognormal distribution, *g(x)* is the second. $\alpha$ and $\beta$ are weighting coefficients for the cumulative probability density of *F(x)*. It shall has the relation $\alpha+\beta=1$.

$$F(x) = \alpha f(x) + \beta g(x) \quad (7)$$

The determination of $\alpha$ and $\beta$ was accomplished by calculating from the ratio of each distribution. $T$ is the total reported number of each distribution; $S$ is the reported number of the first distribution; and $R$ is the reported number of the second distribution. $\alpha$ and $\beta$ are presented in Eqs. (8) and (9), respectively.

$$\alpha = S / T \quad (8)$$

$$\beta = R / T \quad (9)$$

Note that we used two values of $\alpha$ and $\beta$ because two distributions were targeted this time. Hence, multiple distributions can be handled by considering them like that in Eq. (10). $N$ is the number of coexisting distributions, and $c$ denotes the weighting coefficients for the cumulative probability density. In this case, $c_1$ is $\alpha$, $c_2$ is $\beta$, $f_1(x)$ is $f(x)$ and $f_2(x)$ is $g(x)$. In addition, $c$ has the condition of Eq. (11).

$$F(x) = \sum_i^N c_i * f_i(x) \quad (10)$$

$$\sum_i^N c_i = 1 \quad (11)$$

The basic formula of Eq. (7) only strictly represents the probability distribution. Thus, the weighting coefficient must be further determined to represent the access distribution. The access distribution is represented by Eq. (12):

$$H(x) = A * \alpha f(x) + B * \beta g(x) \quad (12)$$

where $A$ is the first peak adjustment coefficient and $B$ is the second peak adjustment coefficient. Similar to that in Eq. (10), handling multiple distributions is possible by considering them like that in Eq. (13). $D$ is the weighting coefficient to represent the access distribution. In this case, $D_1$ is $A$ and $D_2$ is $B$.

$$H(x) = \sum_{i}^{N} D_i * c_i * f_i(x) \quad (13)$$

Substituting $x\_1$ and $x\_2$ of the first and second peaks, respectively, into Eq. (12), and solving the simultaneous equations enables $A$ and $B$ to be determined.

# 4 IMPLEMENTATION AND EVALUATION

## 4.1 Implementation

Table 2 shows the cloud vendor and the instance type to be used in the proposed system. Table 3 shows the cloud API of each vendor and each of the control servers of each vendor. Access to cloud resources is carried out using this API, and it also controls other vendors not only its own vendor. Table 4 shows each server in the system environment. Zabbix is run on each vendor's control server to perform fault detection. For example, if a fault is detected on DC_1 of vendor A, vendor B or C can change the access destination to DC_2 using the aws-cli by changing the setting of the load balancer of vendor A. The web server of the safety confirmation system has machine images of the source code and the OS with the same content. Thus, during failure or scale-out, the machine image is started under the load balancer using the API in Table 3. The machine image is an image of the activation information of the server that includes the middleware (the database management system, etc.), the binary code of the application software, device drivers and OS, and so on. Servers with the same configuration can be rapidly duplicated using a machine image. The Cassandra node also maintains a machine image in the same manner as the web server. At this point, safety information data are not included in the machine image. Cassandra is not necessary at the same time as the machine image safety information data; it only performs data synchronization connected to the cluster with the participation at the time of start-up. Cassandra becomes operational when ready after data synchronization; the status becomes "Up Normal (UN)," indicating normal operation.

## 4.2 Evaluation of Distributed Data Management

In the evaluation of the distributed data management in Cassandra, it was confirmed that the safety information data can be retrieved correctly in an environment that has a stopped DC_1 single node from vendor A. In addition, vendor A was intentionally stopped, and alternative operation by vendors B and C at the backup sites was confirmed.

Table 2: Cloud vendors and instance types

| | | |
|---|---|---|
| Vendor_A | AWS | t2.small |
| Vendor_B | Azure | Standard A1 |
| Vendor_C | Cloudn | Plan v1 |

Table 3: Cloud API

| | |
|---|---|
| AWS | aws-cli 1.10.56 |
| Azure | Azure cli 0.10.3 |
| Cloudn | Cloudn SDK for Ruby 0.0.1 |

Table 4: System environment

| | |
|---|---|
| Safety Confirmation System Web Server | CentOS 6.5<br>Apache 2.2.15 |
| Safety Confirmation System DB Server | CentOS 6.5<br>Cassandra 2.0.6 |
| Control Server | CentOS 6.5<br>Zabbix 2.4.7 |

This is because other nodes also hold the data. As shown in Fig.10, Cassandra copies data to other nodes when data are written to a node from the client. Thus, even when one node stops, the system can operate on other nodes. Therefore, we confirmed that the system can operate with other nodes when a node is stopped. The availability improvement of the proposed system was also confirmed in this manner. Moreover, after the node was stopped, it was confirmed that safety information data are correctly acquired by adding a new node. Node addition and preparation took about 90 seconds. Figure 11 gives a performance comparison of RDB and Cassandra. The evaluation environment is RDB and Cassandra is one respectively, and 12 unit web servers were placed under the DC_1 load balancer of vendor A. We conducted numerous safety report accesses using JMeter to evaluate the environment. The RDB used was PostgreSQL8.4.12. We measured the CPU utilization and throughput of the web server and the DB server in the case where the safety report access increased from zero to 5,000 per 10 minutes. Measurement of CPU utilization was achieved using the "sar" command. Measurement of the throughput was achieved using JMeter. Although CPU utilization of Cassandra is slightly lower in the throughput value of the same degree,

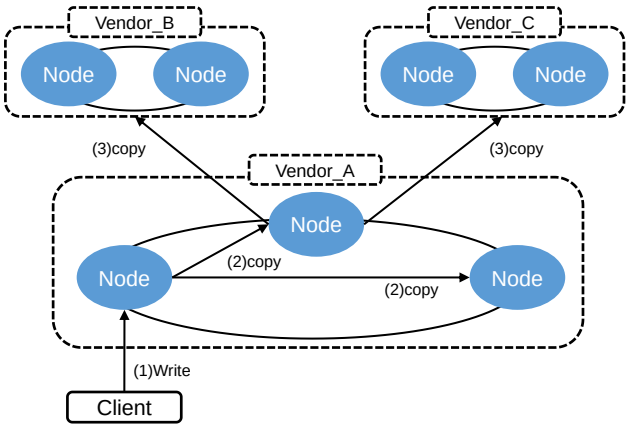


Figure 10: Flow of the data copy operation

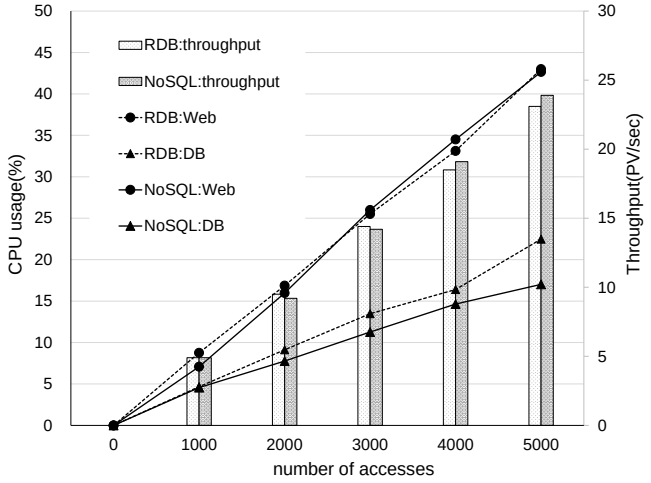


Figure 11: Performance comparison of RDB and NOSQL

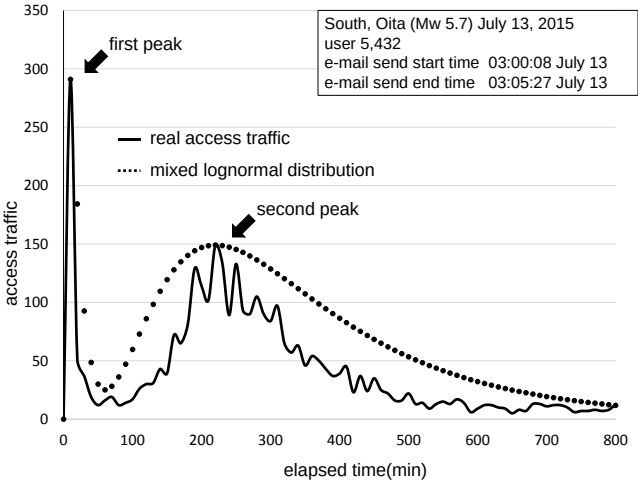


Figure 12: Oita earthquake

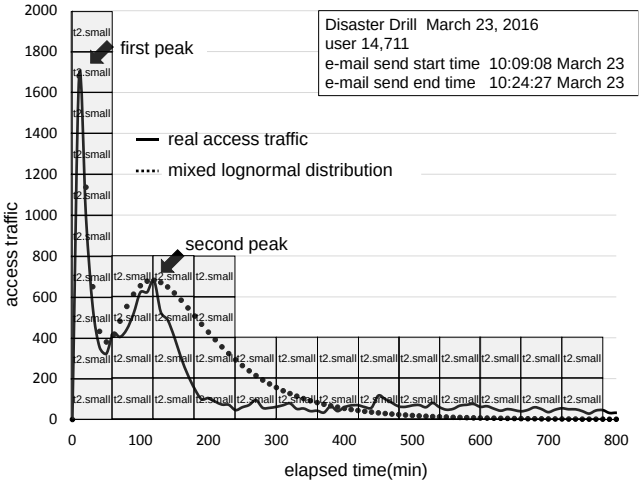


Figure 13: Disaster drill

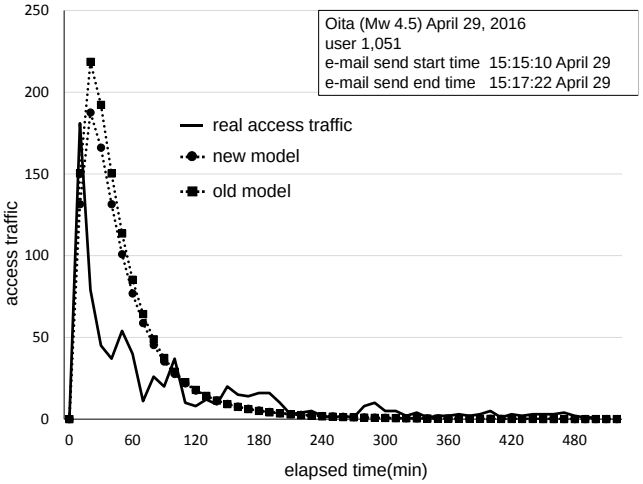


Figure 14: Accuracy of the model during Oita disaster

as the number of accesses increase, there is virtually no difference between both Cassandra and RDB. However, Cassandra is superior in terms of availability using multiple nodes compared with the RDB because Cassandra usually does not operate in a single node.

## 4.3 Evaluation of Suitability for Access Distribution

The suitability of the proposed method was evaluated for access distribution. The evaluation was carried out using 10 minutes of access distribution during the Oita earthquake, shown in Fig.9 (Fig.12), and the disaster drill in a company (Fig.13). A safety confirmation system has a similar load point in a disaster and a disaster drill because access is concentrated from the start of the disaster drill. Therefore, disaster drill data were also used to evaluate the proposed method. Moreover, the disaster drill is similar to the access distribution at midnight. During the disaster drill, the first peak occurred after the start of the drill when the notification e-mail was sent in the morning, and there was a second peak during the lunch break. The number of target users of the Oita earthquake (Fig.12) was 5,432 people. At the first peak, 291 users accessed the system 10 minutes after the occurrence, and the second peak resulted from 149 users accessing the system 220 minutes after the occurrence. Substituting $x\_1$=10, $x\_2$=220 into Eq. (12), and solving the simultaneous equations produces the following results $A$=50,443.9, $B$=64,225.3. Then, $\sigma$ and $\mu$ are determined from $M$ and $E$, for the first and second distribution, respectively, which become the distribution curve of the proposed method, as shown in Fig.12. The number of target users of the disaster drill (Fig.13) is 14,711 people, and the first peak occurs when 1,680 users access the system 10 minutes after the occurrence and the second peak when 681 users access the system 120 minutes after the occurrence. Substituting $x\_1$=10 and $x\_2$=120 into Eq. (12) and solving the simultaneous equations produces the result $A$=154,997.0, $B$=175,775.9. The result shows that the peak of the proposed method is consistent with the access distribution because the known peak value of the past disaster is fitted to Eq. (12). However, the proposed method is also generally acceptable for subsequent distribution. This paper evaluated the suitability of Fig.9 based on the mixed lognormal distribution case having two peaks. However, it is necessary to evaluate with many cases in the future.

It shows the calculation of the number of servers for the calculated access distribution using the proposed method. The number of servers was calculated based on the pro-

cessing power required to access one server with an access distribution of 1-hour increments, because EC2 is charged on an hourly basis. Figure 13 shows that there is access of up to 1,680/10 minutes during 0–60 minutes. The processing power of t2.small is 200 safety report accesses in10 minutes; therefore, 0–60 minutes is for 10 servers. We also calculated the number of servers in the same manner.

### 4.4 Accuracy of the Model by the Number of Sample Data

Each of the parameters used in the proposed model are determined by statistical analysis using data from past disaster and disaster drill data. As an example, *E* is determined by the approximation equation using the relation of the *TU* and *D*. *D* is difference between *E* and *M*. The coefficients to be granted to the lognormal distribution are calculated from the ratio of *TU* and the number of peaks. Therefore, a large amount of data for use in the analysis is expected to improve the accuracy of the proposed method. The access prediction model of a single lognormal distribution was a comparative evaluation of the amount of data, which is less in the old model and more in the new model. The old model was used until the summer of 2015, the amount of data was 14, and the amount of data in the new model to which data was subsequently added, was 29. Figure 14 shows a trace of the disaster data, whereas Fig.15 shows the trace of the disaster drill data. Compared to the old model, both new models are well suited for real access distribution, and it is seen that the accuracy of the new model is improved. The proposed method uses mixed lognormal distribution, which only fits the access distribution to disaster data and disaster drill data. However, as shown with the single lognormal distribution model in Fig.14 and 15, hereafter, data can grasp the tendency of each of the parameters to be given to the model by collecting, and can be expected to build an access prediction model.

## 5 CONCLUSION

In this paper, we proposed a distributed database system that uses multiple servers to improve the availability of safety confirmation systems and an access prediction method that uses a lognormal distribution to predict the concentration of access to the safety confirmation system during a disaster. The DDS using a plurality of Cassandra nodes achieved high availability to continue the operation even when a single node has stopped. The proposed method uses a mixed lognormal distribution and indicates that it is possible to compute access prediction for an access distribution with two peaks resulting from the occurrence of a disaster situation.

Future challenges include the construction of the access prediction model using a mixed lognormal distribution and improving the accuracy. The mixed lognormal distribution model showed the suitability of the extent to which access distribution was allowed during the past disaster and disaster drill. However, this is a poor basis for relevance because the amount of sample data for access prediction was small. A parameter of the model is determined based on past

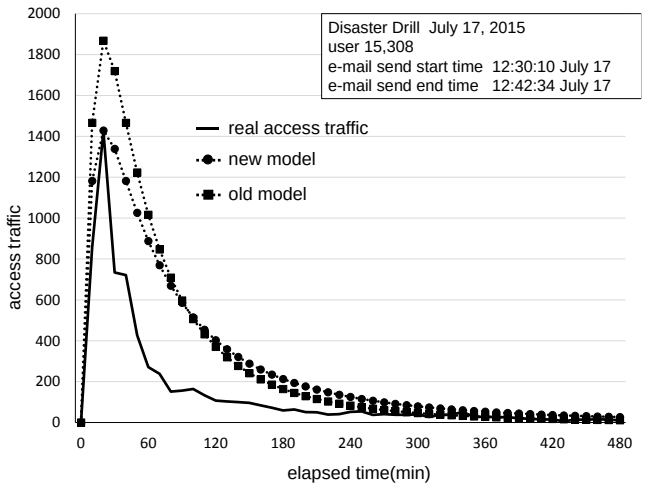


Figure 15: Accuracy of the model during disaster drill

empirical rule with simple consideration. However, using the least squares method or the maximizing likelihood method, it can be expected to further improve the model accuracy. In addition, there is a need for verification and evaluation of the Poisson distribution as well as the normal distribution. In the future, we plan to improve the accuracy and modifications of the proposed method by collecting an additional amount of disaster data.

## ACKNOWLEDGMENTS

We are grateful to Associate Professor Takahiro Hasegawa at the Center for Information Infrastructure, Shizuoka University who developed the safety confirmation system that became the basis for the proposed system. In addition, we are grateful to Kunihiko Murayama, president of AvanceSystem Corporation.

## REFERENCES

[1] T. Hasegawa, H. Inoue, and N. Yamaki, “Development of a low running cost and user friendly safety information system,” Journal for Academic Computing and Networking, No. 13, pp. 91–98 (2009). (in Japanese).
[2] Google Person Finder, https://google.org/personfinder/global/home.html?lang=en, (2016).
[3] J-anpi, http://anpi.jp/top, (2016) (in Japanese).
[4] S. Fu, “Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing,” IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 372–379 (2009).
[5] S. Kajita, Y. Ohta, S. Wakamatsu, and K. Mase, “Stepwise Development of a Survivor Confirmation System for a Higher Educational Institution and Its Production Use,” IPSJ Journal, Vol.49, No.3, pp.1131–1143 (2008). (in Japanese).
[6] H. Echigo and Y. Shibata, “Performance Evaluation of Large Scale Disaster Information System over Japan Gigabit Network,” IEEE 22nd International Conference on Advanced Information Networking and Applications, pp. 1101-1106 (2008).

[7] M. Nagata, Y. Abe, I. Kinpara, M. Fukui, and H. Mineno, “A proposal and evaluation of a global redundant safety information system based on access prediction model,” IPSJ Transactions on Consumer Devices & Systems, Vol. 6, No. 1, pp. 94–105 (2016). (in Japanese).
[8] T. Ishida, A. Sakuraba, K. Sugita, N. Uchida, and Y. Shibata, “Construction of safety confirmation system in the disaster countermeasures headquarters,” Eighth International Conference on 3PGCIC. IEEE, pp. 574–577 (2013).
[9] J. Wang, Z. Cheng, I. Nishiyama, and Y. Zhou, “Design of a safety confirmation system integrating wireless sensor network and smart phones for disaster,” IEEE 6th International Symposium on Embedded Multicore Socs, pp. 139–143 (2012).
[10] H. Yuze, and N. Suzuki, “Development of cloud based safety confirmation system for great disaster,” IEEE 26th International Conference on Advanced Information Networking and Applications Workshops, pp. 1069–1074 (2012).
[11] H. Echigo, H. Yuze, T. Hoshikawa, K. Takahata, N. Sawano, and Y. Shibata, “Robust and large scale distributed disaster information system over internet and Japan Gigabit Network,” IEEE 21st International Conference on Advanced Information Networking and Applications, pp. 762–768 (2007).
[12] Amazon Web Services (AWS), https://aws.amazon.com/?nc1=h_ls, (2016).
[13] Microsoft Azure, https://azure.microsoft.com/, (2016).
[14] Cloudn, http://www.ntt.com/business/services/cloud/iaas/cloudn.html, (2016).
[15] Zabbix, http://www.zabbix.com/, (2016).
[16] JMeter, http://jmeter.apache.org/, (2016).
[17] Amazon EC2, https://aws.amazon.com/ec2/?nc1=h_ls, (2016).
[18] C.D. Murta and G.N. Dutra, “Modeling HTTP service times,” IEEE Global Telecommunications Conference, GLOBECOM'04, Vol. 2, pp. 972–976 (2004).
[19] UnixBench, https://github.com/kdlucas/byte-unixbench, (2016).
[20] Cassandra, http://cassandra.apache.org/, (2016).
[21] N. Matsuura, M. Ohata, K. Ohta, H. Inamura, T. Mizuno, and H. Mineno, “A Proposal of the Distributed Data Management System for Large-scale Sensor Data,” IPSJ Journal, Vol.54, No.2, pp.721–729 (2013). (in Japanese).
[22] F. Toriumi, K. Shinoda, T. Sakaki, K. Kazama, S. Kurihara, and I. Noda, “Analysis of retweet under the Great East Japan Earthquake,” IPSJ SIG Technical Report, Vol. 2012-ICS-168, No. 3, pp. 1–6 (2012). (in Japanese).
[23] Twitter, https://twitter.com/, (2016).

(Received October 7, 2016)
(Reviced June 16, 2017)

**Masaki Nagata**
received his M.E. and Ph.D. degrees from Shizuoka University in 2012 and 2017, respectively. He works for the ASP Division of Avance System Corporation in Hamamatsu city, Japan. He is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”. He also belongs to the Center for Information Infrastructure, Shizuoka University. He is also a member of IPSJ and IEICE.

**Yusuke Abe**
received his B.S. and M.S. degrees from Shizuoka University in 2007 and 2009, respectively. He works for the ASP Division of Avance System Corporation in Hamamatsu city, Japan. He is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”.

**Misato Fukui**
She works for the ASP Division of Avance System Corporation in Hamamatsu city, japan. She is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”.

**Chihiro Isobe**
She works for the ASP Division of Avance System Corporation in Hamamatsu city, japan. She is engaged in web system development such as safety confirmation system “ANPIC” and educational ICT system “SACASS”.

**Hiroshi Mineno**
received his B.E. and M.E. degrees from Shizuoka University in 1997 and 1999, respectively. In 2006, he received his Ph.D. degree from the information science and electrical engineering of Kyushu University. Between 1999 and 2002, he was a researcher of the NTT Service Integration Laboratories. Currently, he is an Associate Professor in the Department of Computer Science of Shizuoka University. His research interests include intelligent IoT system as well as heterogeneous network convergence. He is also a member of ACM, IEICE, IPSJ, and the Informatics Society.