PAPER

Placement of Virtual Storages for Distributed Robust Cloud Storage

Yuya TARUTANI[†a)], **Yuichi OHSITA**[††b)], ***Members, and*** **Masayuki MURATA**[††c)], ***Fellow***

SUMMARY Cloud storage has become popular and is being used to hold important data. As a result, availability becomes important; cloud storage providers should allow users to upload or download data even if some part of the system fails. In this paper, we discuss the distributed cloud storage robust against failures. In the distributed cloud storage, multiple replicas of each data chunk are stored in the virtual storage at geographically different locations. Thus, even if one of the virtual storage systems becomes unavailable, users can access the data chunk from another virtual storage system. In the distributed cloud storage, the placement of the virtual storage system is important; if the placement of the virtual cloud storage system causes that a large number of virtual storages are possible to become unavailable by a failure, a large number of replicas for each data chunk should be prepared to keep the availability. In this paper, we propose a virtual storage placement method that assures availability with a small number of replicas. We evaluated our method by comparing it with three methods. The evaluation shows that our method can keep the availability with 60 % of the network costs required by the comparing method.

key words: *Data Center; Cloud Storage System; Fault tolerance; Redundancy;*

1. Introduction

Cloud storage services have become popular, and a large amount of data is stored in them [1-4]. The cloud storage services are provided via the datacenters; a part of the storage space of the data center is allocated to each user. Then, the users can upload or download their data by accessing the data center. The cloud storage service enables the users to access their data regardless of devices, tools and areas. In addition, by using the cloud storage services, the users can save the cost required to manage the storage devices. Due to the above advantages, the cloud storage services have become used not only by personal users but also by companies.

The availability is important for the cloud storage [5-7] especially used by the companies; the data should always be able to be accessed by the user. However, the storage in a data center may fail. In addition, a data center may become unreachable from the users due to the failure or congestion of the network. Therefore, the service provider of the cloud storage service should keep the availability even in such cases.

One approach to keeping the accessibility to the data even in such cases is to prepare the replicas of the data. The distributed file systems such as Google file system (GFS) [8] or Hadoop file system [9] use this approach to keep the availability of the data. The distributed file systems divide the data into chunks, and store them in one of the storages. At the same time, the replicas of each chunk is stored by the different storages. By doing so, we can access the data even when several storages become unavailable. By using this approach so as to store the replicas in the different data centers, we can keep the availability even if several data centers fail.

Though the approach of preparing replicas keeps availability of the data if a sufficient number of replicas are prepared, more storages or more bandwidth are consumed as the number of replicas increases. The required number of replicas depends on the locations of the data centers storing the replicas; a small number of replicas are sufficient if any possible failure in the network never causes the multiple unreachable data centers storing the replicas.

There are several researches that solves the placement of the data and the backup at the network including multiple data centers [10-12, 5]. This approach places the complete backup of the original data at one of the data centers, so that the backup can be accessed by the user in case that the data center storing original data becomes unreachable. However, this approach requires a large overhead; a large storage resource at a data center may be required to store the complete backup of the data. If one of the data center fails and additional data center storing the backup becomes required, all of the original data should be sent to the additional data center.

In this paper, we discuss the distributed cloud storage that keep the availability of the data even in case of failures without a large number of replicas. This method deploys the *virtual storages* at multiple data centers for each user who requires the availability even in case of failure. In the distributed cloud storage, the data is divided into chunks similar to the existing distributed file systems. Then, by storing each chunk at multiple virtual storages, we keep the availability in

[†]Cybermedia Center, Osaka University, Toyonaka-shi, 560-0043, Japan
[††]Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871, Japan
a) E-mail: y-tarutn@cmc.osaka-u.ac.jp
b) E-mail: y-ohsita@ist.osaka-u.ac.jp
c) E-mail: murata@ist.osaka-u.ac.jp
DOI: 10.1587/trans.E0.??.1

case of failures. This approach makes the placement of the virtual storage flexible; each virtual storage stores only a part of the data. This also reduces the overhead required to place the additional virtual storage, because the size of the virtual storage is not large.

In this paper, we also propose a method to determine the data centers hosting the virtual storages so as to keep the availability even in the case of any possible failures with a small number of replicas. In this method, we use the *split groups*, which is defined as the set of groups including the nodes belonging to the same connected subgraph if the failure occurs. The nodes belonging to the same split group can communicate with each other even if the failure occurs. Thus, our method guarantees the availability of the data by placing the virtual storages so that the split groups including the node connected to the user includes more $N-k$ virtual storages, where N is the number of deployed virtual storages and k is the number of replicas.

Our method determines the data center hosting the virtual storages and the number of replicas by searching the suitable set of the data centers for the small number of replicas. When we cannot find the suitable set of the data centers, we increment the number of replicas, and search the suitable set again. By continuing these steps, we determine the data centers hosting the virtual storages that keep the availability with a small number of replicas.

The rest of this paper is organized as follows. In Section 2, we explain a cloud storage system discussed in this paper. Section 3 presents a heuristic method for deciding the data centers hosting the virtual storages. In Section 4, we evaluate our method by comparing with a method without considering the required number of replicas. Finally, Section 5 provides our conclusions.

2. Distributed clouds storage

Figure 1 shows the overview of the distributed cloud storage discussed in this paper. The distributed cloud storage discussed in this paper is constructed of multiple data centers and the network between them. The service provider of the storage places multiple *virtual storages* for each user. Each virtual storage is hosted by one of the data centers by using a part of the storage of the data center. The set of the virtual storages stores the user's data.

The data stored in the distributed cloud storage is divided into multiple small chunks, and each chunk is stored in k_u virtual storages. By doing so, we keep all fraction of data unless all of k_u virtual storages storing the chunk become unavailable simultaneously. Each chunk has the ID, by which the virtual storage storing it is determined. By using the k_u hash functions, we can determine the set of the virtual storages storing the chunk.

For reducing the communication delay from users and the amount of traffic on networks, the virtual storages are stored in the data centers that are close to users. However, the number of unavailable virtual storage becomes large in the case of failure, because most of virtual storages are concentrated in the data centers that are close to users. This causes an increase in recovery time and network overhead for recovering the robustness. In this paper, the data centers hosting the virtual storages are determined on the basis of an impact of failures to reduce the recovery time and the network overhead for recovering the robustness.

In the rest of this section, we explain the operation on the distributed cloud storage.

2.1 Access from users

The user accesses the data on the distributed cloud storage through a client software. The client software knows the hash functions, by which the client obtains the virtual storage storing the chunk.

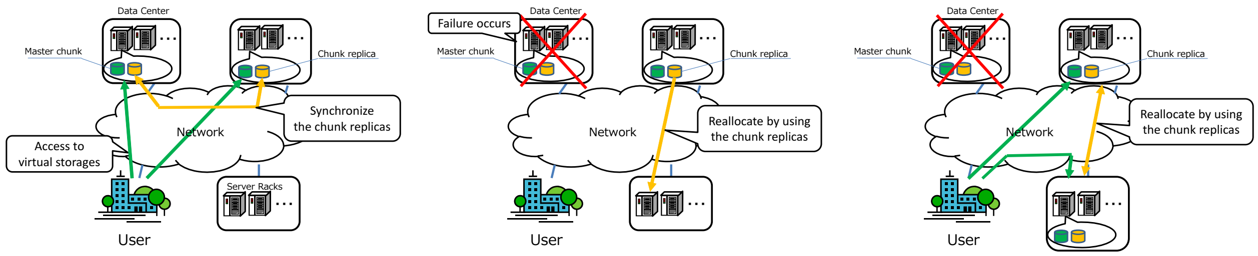
Data is downloaded by downloading all the chunks required to construct the data. For each chunk, the client software calculates the hash functions to obtain the lists of the virtual storages storing the chunk. Then, the client software selects one of them, and downloads the chunk from the selected one.

Similarly, when uploading a data, the client software uploads all the chunks included in the data. Each chunk is uploaded to one of the virtual storages corresponding to the ID of the chunk, calculated by the hash functions. Then the virtual storage copies the chunk to the other virtual storages corresponding to the ID of the chunk.

2.2 Management of the set of the virtual storages

The set of the virtual storages for each user is managed by the central manager. The central manager knows the location of the users and data centers in the network, the remaining resources of the data centers, available bandwidth of the links in the network, and the set of nodes or links that are possible to fail simultaneously. Based on this information, the central manager controls the locations of the virtual storages so as to keep the availability even in case of the failures.

After placing the virtual storages, the central controller monitors the set of the virtual storages. If the set of the virtual storage does not satisfy the requirements due to the virtual storages that become unavailable, the central controller detects it, and new virtual storages are placed. The chunks stored in the new virtual storages are obtained from the other active virtual storages; the virtual storages storing the chunks required to be stored by the new virtual storages can be obtained by the hash functions, and then the new virtual storages send a request to the active virtual storages.



(a) Normal condition (b) Failure occurs (c) Repairing the Cloud Storage

Fig. 1 Distributed Robust Cloud Storage

3. Placement of the virtual storages

In this paper, the central controller decides the data centers hosting the virtual storages for each user. The placement of the virtual storages have an impact on the number of required replicas. Because the required storages and the bandwidth between data centers increase as the number of replicas becomes large, the central controller should determine the data centers hosting the virtual storages so as to keep the availability without a large number of replicas, and then determine the number of replicas. In this section, we formulate the problem to decide the data center hosting the virtual storages, and propose a heuristic method to solve the problem.

3.1 Problem Formulation

3.1.1 Input

The central controller knows the network topology. The network is represented as a graph G. We denote the set of nodes by N and the set of edges by L. Each edge l has available bandwidth b_l^{avail}. There are multiple data centers that can host the virtual storages, and all of the data center is connected to the network. We denote the set of data center by D. We denote the storage space provided by the data center $d \in D$ by p_d. Each user is also connected to at least one of the nodes in the network. We denote the set of users by U, and the set of the nodes connected to the user u by N_u.

A path on the graph G can be represented as a set of the links on the path. There may be multiple paths between any two nodes. We denote the set of paths between the nodes a and node b by $r_{a,b}$, and the set of all paths on the graph G by R. Among the paths included in $r_{a,b}$, we denote the shortest one by $r_{a,b}^{\text{short}}$.

The central controller also knows the possible patterns of failures. The set of links that are possible to fail simultaneously is called the shared risk group (SRG). We denote the set of the SRGs by F, and the set of the nodes included in the SRG f by N_f^{fail}. The central controller receives the requests from the user. The request includes the number of required virtual storages C_u, the traffic rate from the user uploading data b_u^{upload} and the traffic rate to the user downloading data b_u^{download}.

3.1.2 Variables

We determine the data centers hosting the virtual storages for the user u by setting a variable $M_{n,u}$ which indicates the number of virtual storages for the user u hosted by the node n. In this problem, in addition to the data centers hosting the virtual storages, we also determine the number of replicas. We denoted the number of replicas for the user u by k_u.

In the rest of this paper, we set the size of each virtual storage to 1 in order to make discussion easy. As a result, the total number of virtual storages for the user u is $(1+k_u)C_u$. In addition to the above variables, we also define a variable b_l indicating the traffic amount of the distributed cloud storage on the link l.

3.1.3 Objective

In this paper, we minimize the network cost defined by the total bandwidth used by the distributed cloud storage.

$$\text{minimize} \sum_{l \in L} b_l. \tag{1}$$

3.1.4 Constraints

- The total number of virtual storages hosted by data centers should be $(1 + k_u)C_u$

$$\forall u \in U\colon \sum_{n \in D} M_{n,u} = (1 + k_u)C_u.$$

- All the data centers must have the sufficient resources to host all of the allocated chunks, including chunk replicas.

$$\forall n \in D\colon \sum_{u \in U} M_{n,u} \leq U_n,$$

where U_n is the maximum number of virtual storage in data center n.

- All data must be available even if any failure occurs. Because each chunk is stored by k_u+1 virtual storages, all data is available unless more than k_u virtual storages become unreachable from the user u. That is,

$$\forall u \in U, \forall f \in F\colon\ N^{\text{unreach}}_{u,f} \leq k_u,$$

where $N^{\text{unreach}}_{u,f}$ is the number of virtual storages that become unreachable from the user u when the set of nodes is included in N^{fail}_f, which is calculated by

$$N^{\text{unreach}}_{u,f} = \sum_{n\in\{n|n\in D, \forall r\in r_{N_u,n}, \exists m\in N^{\text{fail}}_f : m\in r\}} M_{n,u}$$

- b_l is the sum of the traffic passing the link l. In this distributed cloud storage, data is divided into multiple small chunks, and chunks are stored in separate virtual storages. Therefore, we assume that the users access all virtual storages at the same rate. The traffic rate from the user u to each virtual storage is denoted by $\frac{b^{\text{upload}}_u}{C_u}$ where $b^{\text{upload}}_u istheamountoftraffic$ at the worst case. Similarly the traffic rate from each virtual storage to the user u is denoted by $\frac{b^{\text{download}}_u}{C_u}$ where $b^{\text{download}}_u istheamountoftraffic$ at the worst case. Each uploaded chunk is copied to k_u virtual storages. In this paper, we assume that the virtual storage hosting a replica is selected at the same probability, because the calculation overhead for solving the placements of replicas is avoided. This problem is considered in future work. Thus, the traffic rate between the virtual storages of the user u is $\frac{b^{\text{upload}}_u}{C_u}$. That is,

$$b_l = \sum_{u\in U, n\in D, l\in r^{\text{short}}_{N_u,n}} \left(\frac{M_{d,u}(b^{\text{upload}}_u + b^{\text{download}}_u)}{C_u}\right) + \sum_{u\in U, n_1,n_2\in D, l\in r^{\text{short}}_{n_1,n_2}} \left(\frac{M_{n_1,u}M_{n_2,u}b^{\text{upload}}_u}{C_u}\right)$$

- b_l should be less than the available bandwidth of the link l

$$\forall l \in L\colon\ b_l \leq b^{\text{avail}}_l.$$

3.2 Heuristic Method to Place the Virtual Storages

Solving optimization problem formulated in Section 3.1 requires a large time, because the problem includes 6 integer variables. Therefore, we propose a heuristic method to determine the data centers hosting the virtual storages.

To determine the data centers hosting the virtual storages, we use the *split groups*, which is defined as the set of groups including the nodes belonging to the same connected subgraph if the failure occurs. The nodes belonging to the same split group can communicate with each other even if the failure occurs. Thus, we can guarantee the availability of the data by placing the virtual storages so that the split groups including the node connected to the user includes more than $|N| - k$ virtual storages.

In our method, the data centers hosting the virtual storages are determined by the following steps. We first calculate the split group for each SRG in advance. Then, we decide the data center hosting the virtual storages one by one by using the split groups. The rest of this section explains the details of the above steps.

3.2.1 Calculation of the Split Groups

The split groups for SRG f is obtained by following steps.

1. Obtain the set of nodes $S_{N^{\text{fail}}_f}$ that is connected to one of the links in N^{fail}_f
2. Construct the graph G' where the links in N^{fail}_f are removed from the network G.
3. Calculate the route from the node $s_{f,1}$ to the node $s_{f,2}$ on the graph G' by using the Dijkstra algorithm, for all node pairs $s_{f,1}$ and $s_{f,2}$ included in $S_{N^{fail}_f}$
4. Construct groups so that the nodes $s_{f,1}$ and $s_{f,2}$ belongs to the same group only when the routes between $s_{f,1}$ and $s_{f,2}$ is found in the previous step.

The calculation time to obtain the split group for each SRG is $\mathcal{O}|N|^2$. In a large-scale network, this calculation time becomes large, because the number of SRG increases. Therefore, our future work is to reduce the calculation time to obtain the split group for each SRG by combining several SRG into one SRG.

3.2.2 Placement of the Virtual Storages

Figure 2 shows the flowchart to determine the data centers hosting the virtual storages for the user u. In these steps, we first set k_u to 1 and search the suitable set of data centers hosting the virtual storages which achieves the availability of the data for all SRGs even when k_u is 1, to minimize the number of required replicas. Then, if we cannot achieve the availability of the data for all SRGs, we increment k_u, and go back to searching the suitable set.

When searching the suitable set of the data centers, we determine the data center hosting virtual storages one by one. The candidate data centers hosting a virtual storage is checked in the ascending order of the

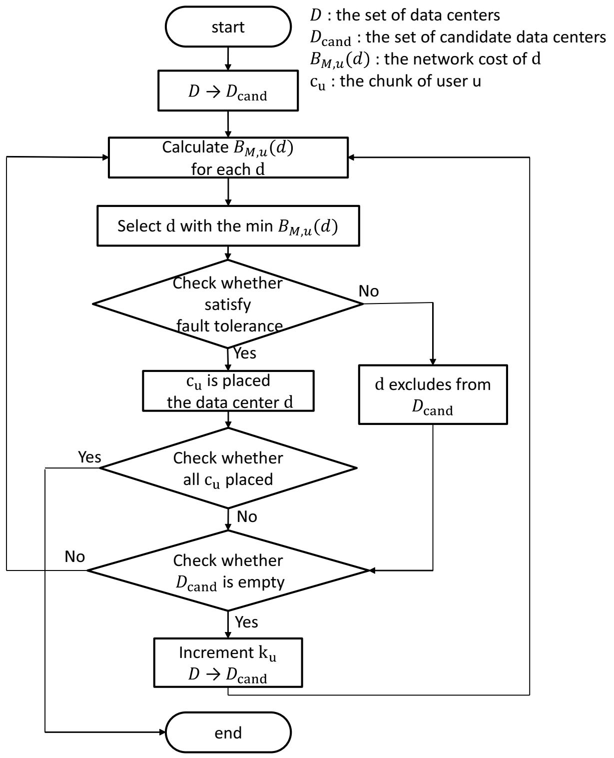


Fig. 2 Flowchart of our method

network cost $B_{M,u}(d)$ caused by the traffic from/to the data center defined by

$$B_{M,u}(d) = (b_u^{\text{upload}} + b_u^{\text{download}})|r_{n_u,d}| + \sum_{n \in D} M_{n,u} k_u b_u^{\text{upload}} |r_{n,d}|. \quad (2)$$

where $M_{n,u}$ is the number of virtual storages whose locations are already decided to the data center n, and n_u is the node corresponding to the user u in the graph G. By selecting the data center with the smallest $B_{M,u}(d)$, we avoid a large network cost.

When checking whether a data center is suitable to hosting the virtual storage, we check whether the constraints of the availability are not violated. We can keep the availability unless more than k_u virtual storages become unreachable from the user. That is, for all of the SRGs in F, the number of the virtual storages hosted by the data centers belonging to different split group from the user should be less than k_u, so as to keep the availability in case of failure.

Therefore, in our method, we count the number of virtual storages hosted by the data centers in different split group from the user for each of the SRGs. Then, if the number exceeds k_u, we regard the selected data center unsuitable data center to host the virtual storage, and eliminate it from the candidate data centers.

The computational complexity for selecting the data centers to host the virtual storage of user u is $\mathcal{O}(|C_u|)$.

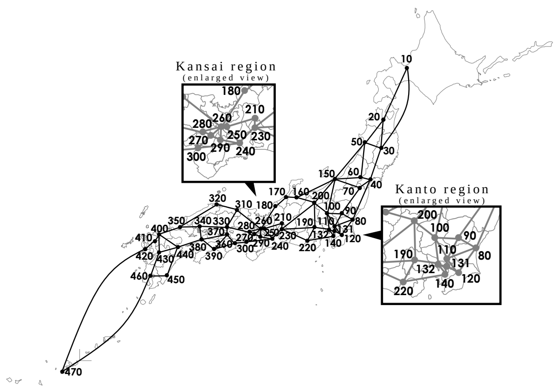


Fig. 3 Japane Photonic Network Model

4. Evaluation

In this section, we evaluate our method, and demonstrate the advantage of the placement of the virtual storages considering the impact of failure.

4.1 Evaluation Environment

4.1.1 Network topology

In this evaluation, we use the Japan Photonic Network Model (JPNM), which is a model of the network in Japan [13] shown in Fig. 3. This network topology is likely to divide the subgraphs by failure. In this case, the impact of failure is depended on the location of the data centers. Therefore, in this evaluation, we place 8 data centers whose location are selected close to users, and far from users. The number of virtual storage of data center is set to 1000, and the bandwidth of link is set to 10G bit per seconds. The size of virtual storage is set to 500GBytes.

In this evaluation, the SRG is created so that the failure divides the network into the 2 subgraphs. The SRG is given by following steps for each data center.

1. Select the closest data center d to a data center s.
2. Add the links on the shortest path between s and d to the SRG.
3. Remove the links on the shortest path between s and d.
4. Check the whether the path exists. If yes, go back to Step 2, otherwise end.

4.1.2 Request from users

We assume that the users are only in Tokyo and Osaka, and the number of users in Tokyo equals to the number of users in Osaka. Each user connected to two nodes in the network, because users can access their virtual storages in the case of failure. We connect the users in Osaka to the nodes 260 and 270, and users in Tokyo to the nodes 131 and 132. The number of users at each location is set to 25. The size of the storages requested by each user is set to an uniform random value from 5 to 20. Each user uploads and downloads 10Mbit per second.

4.1.3 Evaluation metrics

In our evaluation, we investigated the resource required by the distributed cloud storage. The network resource required by the distributed cloud storage is estimated by the network cost defined by Eq. 2. The resource of the data centers is estimated by the size of the storages allocated for the distributed cloud storage, which is calculated by

$$C^{\text{all}} = \sum_{u \in U} C_u(k_u + 1). \qquad (3)$$

Moreover, we evaluate the availability and restoration time for recovering robustness. The availability of data of user u is guaranteed by placing the virtual storage so that the split groups including the node connected to the user includes more than $|C_u| - k_u$ virtual storages. Thus, the availability of data of user u is denoted by following;

$$\text{Availability} = \begin{cases} 100 & (C_u - k_u \leq N^{\text{reach}}_{u,f}) \\ 100\frac{N^{\text{reach}}_{u,f}}{C_u - k_u} & (Otherwise) \end{cases} \qquad (4)$$

where $N^{\text{reach}}_{u,f}$ is the number of virtual storage that become reachable from the user u when the set of nodes included in N^{fail}_f.

The data centers hosting the virtual storages for recovering are also determined by using the method to determine the data center hosting the virtual storages firstly. In the case of restoration, the virtual storages are sent to data centers by using the available bandwidth of links. In this evaluation, we assume that the delay is depended on the amount of sending data, because the transmission delay is much larger than the delay depending on the distance. Thus, the restoration time t_{rest} is denoted by following;

$$t_{\text{rest}} = \frac{V_d}{\min_{l \in r^{\text{short}}_{s,d}} b^{\text{avail}}_l} \qquad (5)$$

where V_d is the amount of sending data and $r^{\text{short}}_{s,d}$ is the shortest path from the source data center s to the restoring data center d.

4.1.4 Compared Method

In this evaluation, we compare with three methods that guarantee the availability of the data by placing the virtual storages. In this paper, we called these method *mirroring method*, *random method*, and *last determining method*, respectively.

The mirroring method is similar to a method described in Ref. [5]. This method guarantees the availability of the data by preparing the backup virtual storages. In this method, the service provider selects the data center hosting the virtual storages within the shortest distance from user. For guaranteeing the availability of the data, another data center hosting the backup virtual storages is also selected. In this paper, the mirroring method selects the data center that users can access their virtual storages or their backup virtual storages in the case of failure.

The random method is similar to GFS [8]. In this method, we determine the data center hosting the virtual storages randomly. Moreover, the number of replicas is a constant value without considering the placement of virtual storages. In this paper, we set the number of replicas to the quarter of required virtual storages.

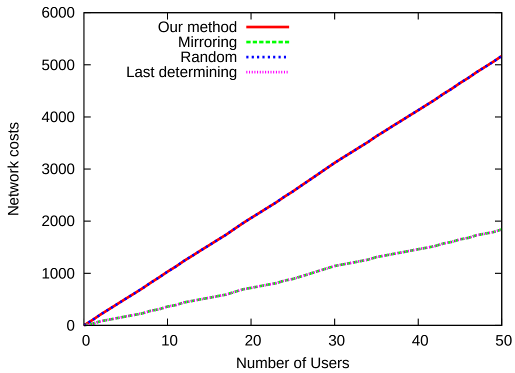
Finally, the last determining method determines the number of replicas after determining the data centers hosting the virtual storages. In this method, we first determine the data centers hosting the virtual storages so that the network cost is minimized. This can be calculated by using the method described in Section 3.2.2 with setting b_u to a sufficiently large value C_u. Then, b_u is set as the minimum value without violating the availability in any case of the SRG. Finally, the number of the virtual storages in each data center is adjusted so as to minimize the used storage size under the constraint that b_u replicas per each chunk can be stored.

The purpose of the last determining method is to minimize the network cost for the user to access the data. On the other hands, this method does not consider that the network cost is increased by synchronizing the replicas. Therefore, our evaluation shows the advantage of determining the placement of virtual storage with considering the number of replica.

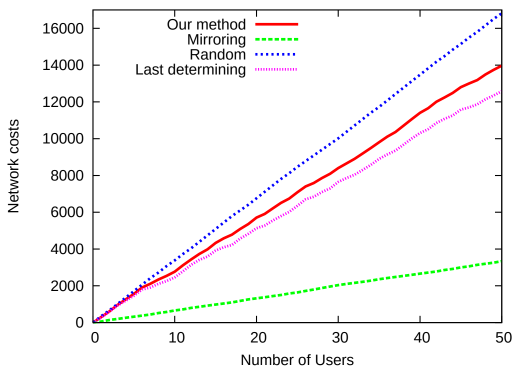
4.2 Results

4.2.1 Cost of networks

Figure 4 shows the network cost of our method and the comparing methods in the case that the data centers are placed far form users. In these figures, the horizontal axis is the number of users, and the vertical axis is the network cost calculated by Eq. (2). Figure 4(a) shows the network cost caused by the communication between users and data centers. This figure indicates that the mirroring method and the last determining method achieve the smaller network cost than our method and random method. This is caused by that the mirroring method and the last determin-

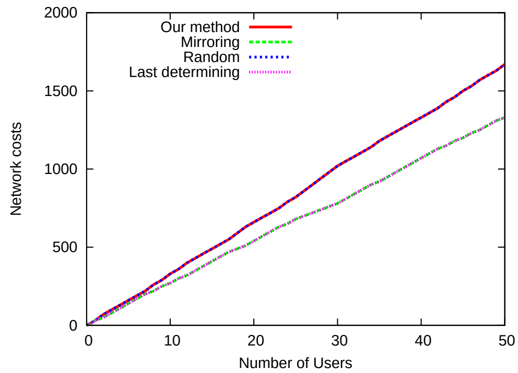


(a) Users

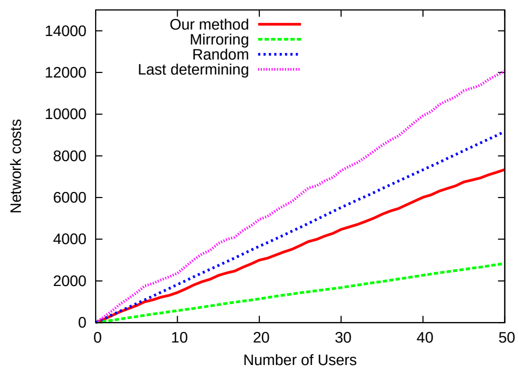


(b) Total

Fig. 4 The cost of network in the case of data center placed far from users



(a) Users



(b) Total

Fig. 5 The cost of network in the case of data center placed near users

ing method deploy the virtual storages near the users, while our method and random method deploy virtual storages also at the data center far from the user to keep the availability in case of failure. As a result, the number of hops from the user to the data centers is large in our method and the random method, compared with the mirroring method and the last determining method.

However, Fig. 4(b) shows that the total cost of our method is similar to that of the last determining method. This is because the last determining method requires more replicas so as to keep the availability. As a result, the data centers send a large amount of data so as to keep the replicas updated when the last determining method is used, which causes the large network cost. On the other hands, Fig. 4(b) also shows that the total cost of the mirroring method is smaller than other methods. This is because, in the mirroring method, the number of backup virtual storages is smaller than other methods. As a result, the amount of traffic for guaranteeing the availability is small.

Figure 5 shows the network cost of our method and the comparing methods in the case that the data centers are placed near users. In these figures, the horizontal axis is the number of users, and the vertical axis is the network cost calculated by Eq. (2). Figure 5(a) indicates that the mirroring method and the last determining method achieve the smaller network cost than our method and random method, but the difference is smaller than the case of Fig. 4. This is because, the placements of data centers are near users. As a result, the number of hops from user to the data centers is small in our method. Therefore, our method is more suitable in this case. Figure 5(b) also shows that the total cost of our method is smaller than the last determining method. This is because, the number of replicas by our method is smaller than the last determining method, and the number of hops between data centers is small. As a result, our method can keep the availability with 60 % of the network costs required by the last determining method.

Figure 6 shows the total number of virtual storages. In this figure, the horizontal axis is the number of

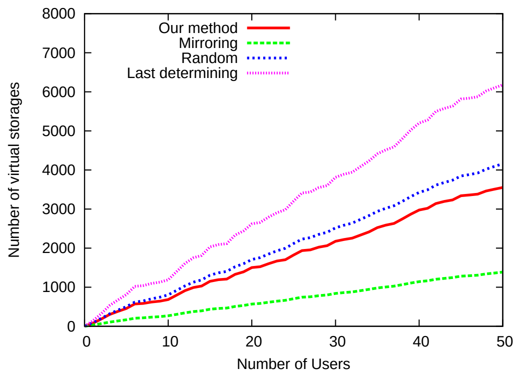


Fig. 6 The Number of virtual storages

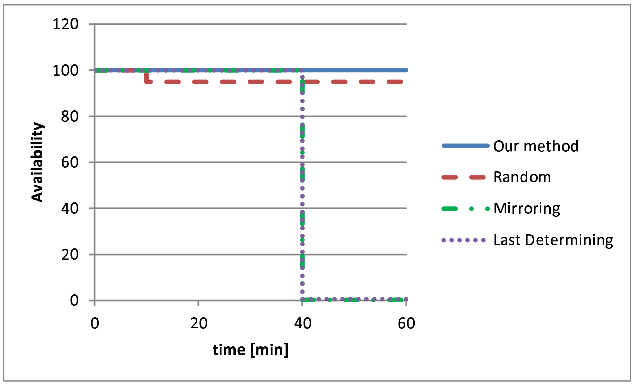


Fig. 7 Availability of data

Table 1 Restoration time

Method	maximum restoration time
Our method	27 minutes
Mirroring	133 minutes
Random	30 minutes
Last Determining	67 minutes

users, and the vertical axis is the total number of virtual storages. Figure 6 shows that the total number of virtual storage used by our method is much smaller than that of the last determining method. This is because our method decides the data centers hosting the virtual storages so that a large number of virtual storages never become unavailable simultaneously. As a result, in our method and random method, a small number of replicas are sufficient to keep the availability in case of failures, which leads to the reduction of the required resources.

4.2.2 Availability

We evaluate the availability in the case of failure. In this evaluation, the SRG selected randomly is fail. The time of fail is 10 minutes and 40 minutes. In this evaluation, the number of cases is set to 20, and the result shows the worst case. Figure 7 shows the result. In this figure, the horizontal axis is the time, and the vertical axis is the availability defined as Eq. (4). Figure 7 indicates that our method, the mirroring method and the last determining method achieve the availability of data in the case of first failure. On the other hands, the random method cannot achieve the availability of data, because the number of replica is small for guaranteeing the availability.

However, the mirroring method and the last determining method cannot achieve the availability of data in the case of second failure. Table 1 shows the time for restoring the robustness. The restoration time of these method is large, because the sending data size is large in these method. Therefore, the mirroring method and the last determining method is not suitable in the case of multiple failure. On the other hands, our method can also achieve the availability of data in this case. This is because, the number of unreachable virtual storage from users is small. As a result, the restoration time is small.

5. Conclusion

In this paper, we discuss the distributed cloud storage that keeps the availability of the data even in case of failure without a large number of replicas. We also propose a method to determine the data centers hosting the virtual storages so as to keep the availability even in the case of any possible failures with a small number of replicas. In our method we use the split-groups, which is defined as the set of groups including the nodes belonging to the same connected subgraph if the failure occurs. In this method, we calculate the split-group for each node, which is the group that the nodes are grouped based on the connectivity when the node fails. We evaluated our method by comparing three methods. The evaluation shows that our method keep the availability with only 60 % of the network costs required by the method without considering the required number of replicas.

In this paper, the placement of chunk replicas is selected randomly. We expect that the number of chunk replicas and the cost of network can be reduced by considering the placement of chunk replicas. Moreover, in our method, the split-group must be recalculated when the failure occurs. Therefore, reducing the calculation time is one of future work in the large-scale network.

Acknowledgment

This work was supported in part by the National Institute of Information and Communications Technology (NICT).

References

[1] "Google Drive." `https://www.google.com/intl/en/drive/`.

[2] "Dropbox." `https://www.dropbox.com/`.
[3] "Microsoft OneDrive." `https://onedrive.live.com/`.
[4] "box." `https://www.box.com/`.
[5] A. Xiao, Y. Wang, L. Meng, X. Qiu, and W. Li, "Topology-aware virtual network embedding to survive multiple node failures," in *Proceedings of GLOBECOM 2014*, pp. 1823–1828, Dec. 2014.
[6] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 187–198, Nov. 2009.
[7] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, *et al.*, "Windows azure storage: a highly available cloud storage service with strong consistency," in *Proceedings of the 23 ACM Symposium on Operating Systems Principles*, pp. 143–157, Oct. 2011.
[8] S. Ghemawat, H. Gobioff, and S. Leung, "The google file system," in *Proceeding of ACM SIGOPS Operating Systems Review*, vol. 37, pp. 29–43, Dec. 2003.
[9] "Apache hadoop project." `http://hadoop.apache.org/`.
[10] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures," in *Proceedings of IEEE GLOBECOM*, pp. 1–6, Dec. 2010.
[11] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive virtual network provisioning," in *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, pp. 41–48, Sept. 2010.
[12] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," in *Proceedings of the ACM SIGCOMM*, pp. 431–442, Aug. 2012.
[13] "Japan Photonic Network Model." `http://www.ieice.org/pn/jpn/jpnm.html`.

Yuya Tarutani received an M.E. and a Ph.D. in Information Science and Technology from Osaka University in 2012 and 2014, respectively.
He is currently an Assistant Professor at the Cybermedia Center, Osaka University. His research interests include traffic matrix estimation, data center networks and network reconfiguration. He is a Member of IEICE and IEEE.

Yuichi Ohsita received an M.E. and a Ph.D. in Information Science and Technology from Osaka University in 2005 and 2008, respectively.
He is currently an Assistant Professor at the Graduate School of Information Science and Technology, Osaka University. His research interests include traffic matrix estimation and countermeasures against DDoS attacks. He is a Member of IEICE, IEEE, and the Association for Computing Machinery.

Masayuki Murata received an M.E. and a D.E. in Information and Computer Sciences from Osaka University in 1984 and 1988, respectively.
In April 1984, he joined the Tokyo Research Laboratory of IBM Japan as a researcher. From September 1987 to January 1989, he was an Assistant Professor at the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate Professor at the Graduate School of Engineering Science, Osaka University, and became a Professor at the same school in April 1999. He moved to the Graduate School of Information Science and Technology, Osaka University in April 2004. He has published more than 300 papers in international and domestic journals, and has given presentations at numerous conferences. His research interests include computer communication networks, as well as performance modeling and evaluation.
He is a Fellow of IEICE and a Member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.