Electronic Preprint for Journal of Information Processing Vol.24 No.2

Regular Paper

Comparison of Distance Limiting Methods for Risk-aware Data Replication in Urban and Suburban Area

Takaki Nakamura[1] Shinya Matsumoto[2] Masaru Tezuka[3] Satoru Izumi[4]
Hiroaki Muraoka[1]

Received: June 14, 2015, Accepted: December 7, 2015

Abstract: Risk-aware data replication (RDR), which replicates data at primary sites to safe backup sites, has been proposed to mitigate a service disruption in a disaster area even after a widespread disaster that damages a network and a primary site. RDR assigns a safe backup site to a primary site while considering a damage risk for both the primary site and the backup candidate site. When the backup candidate sites are widely distributed in an urban and suburban area, RDR sometimes assigns a backup site too far from the primary site. However the backup site is desired to be reachable from the primary site by physical transfer such as walking, bicycle, car, or drone in case that a severe disaster damages network among the sites. Therefore, limiting the distance between the primary site and the backup site is required. To approach this challenge, we propose two possible methods: the average distance limiting (ADL) method and the maximum distance limiting (MDL) method. In this paper, we compare the distance distributions, the data availability, and the computation time of two methods. Then, we conclude that the MDL method is the most practicable from a comprehensive perspective.

Keywords: remote replication, disaster recovery, distributed storage, integer programming problem, availability

1. Introduction

Remote replication [1], [2], [3] is widely used to provide high availability information services after geographically widespread disasters. This feature replicates data on a primary site to a distant backup site. The backup site takes over information services from the primary site once a disaster occurs in the primary site area. This is extremely common yet sophisticated technology. Nevertheless, severe and widespread disasters have shown that remote replication is not sufficient because severe disasters can damage information networks as well as information servers in the disaster area. Therefore, people in the disaster area become isolated from the backup site area via the wide area network including the internet. Under such severe circumstances, it is difficult for existing technologies to sustain the information services in the disaster area.

In fact, in the case of the East Japan Great Earthquake of 2011, disaster victims in severely affected disaster areas were unable to access the internet for a month or longer [4]. They were therefore unable to benefit from information services such as resident registries and medical histories. The resident registries were necessary to identify whether residents were safe or not. Medical history information was necessary to sustain their health immediately after a disaster.

A feasible idea to tackle this social problem is to replicate data at a primary site to a nearby site. However the nearby site might be damaged by a widespread disaster which damages the primary site. Additionally, creating several replicas at nearby sites increases the cost of the storage system. Therefore, risk-aware data replication (RDR), which replicates data to nearby safe sites while considering the damage risk at both primary sites and backup sites, has been proposed. If the data survives in the nearby area, it will become accessible via either a local area network or by transferring to/from the network-isolated backup site having the surviving storage apparatus. In previous studies [5], [6], formulation of the Integer Programming Problem (IPP) for RDR was shown. Improved results of data availability were confirmed by disaster simulations on a virtual urban-sized field.

Extending the field to an urban and suburban area raises an issue that RDR sometimes assigns a backup site too far from the primary site. This distance becomes a physical barrier that prevents accessing the surviving storage apparatus on a network-isolated backup site.

To approach this issue, we propose two possible methods: the average distance limiting (ADL) method and the maximum distance limiting (MDL) method. The ADL method limits the average distance of primary-backup site pairs. The MDL method limits the maximum distance of primary-backup site pairs. The ADL method is expected to give a higher data availability at the expense of computation time. The MDL method is expected to give a fast solution at the risk of lower data availability. In this paper, we compare these two methods from the perspectives of the

[1] Research Institute of Electrical Communication, Tohoku University, Sendai, Miyagi 980–8577, Japan
[2] Research & Development Group, Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[3] Hitachi Solutions East Japan, Ltd., Sendai, Miyagi 980–0014, Japan
[4] Cyberscience Center, Tohoku University, Sendai, Miyagi 980–8577, Japan

distance distributions, the data availability, and the computation time.

The remainder of this paper is organized as follows. We explain RDR and the issue to apply RDR to an urban and suburban area in Section 2. In Section 3, two distance limiting methods: the ADL method and the MDL method are proposed. Then constraints and a revised objective function of IPP for each method are shown. In Section 4, simulation conditions are described. In Section 5, we compare the distance distributions, the data availability, and the computation time of two proposed methods. We present related work and conclusions in Sections 6 and 7, respectively.

2. Risk-aware Data Replication and Issues

2.1 Overview of Risk-aware Data Replication

Risk-aware data replication (RDR) replicates data at a primary site to a nearby safe site while considering the damage risks for both the primary site and the backup site. **Figure 1** shows a conceptual diagram of RDR. Four sites exist in the urban and suburban area, each with an information server. The information server at site S1 in the urban area operates an information service. The three candidate backup sites are sites S2 in the urban area, S3, and S4 in the suburban area. We assume that these areas will be damaged by an earthquake in the near future. Because the lowest risk site for the earthquake damaging site S1 is site S3, the data at site S1 should be replicated to site S3. It is easy to decide the backup site in this case because the number of sites is few. Moreover, only one primary site exists.

2.2 Use Cases of Risk-aware Data Replication

RDR can be applied to not only an earthquake but also a wide variety of natural disasters and artificial disasters if the damage risks can be estimated with a reasonable accuracy. The natural disasters include a tsunami, a flood, a typhoon, a cyclone, a hurricane, a landslide by a heavy rain, an explosion of a volcano, and so on. The artificial disasters include a meltdown of a nuclear reactor, a massive blackout, a synchronized terrorist attack, a war, and so on.

Target data to be protected by RDR is urgently required data after the disasters including resident registries and medical histories. Therefore, target facilities as sites of RDR are city offices, pharmacies, hospitals, emergency evacuation centers, and so on. As for pharmacies, the number of sites is more than 500 in Sendai city and more than 1,100 in Miyagi prefecture.

2.3 Mathematical Model of Risk-aware Data Replication

When the number of primary and backup sites extends beyond the hundreds it becomes difficult to decide the pairs of primary-backup sites manually. We can use the Integer Programing Problem (IPP), which is a mathematical optimization problem for site-pairing of RDR to massively multiple sites. The IPP consists of an objective function and constraints, including integer variables.

A formulation of the IPP is presented below. **Figure 2** shows an example of the relation between the variables describing each site when the number of sites is 4. The relevant variables are the risk indicator P, the unused capacity F, and the used capacity D with an index to distinguish each site. Their detailed definitions are presented in the following subsections.

1) Objective Function

An objective function is described as

$$f(x_{12}, \cdots, x_{n(n-1)}) = \sum_{i=1, i\neq j}^{n} \sum_{j=1}^{n} D_i P_{ij} x_{ij}, \quad (1)$$

where $x_{ij} \in \{0, 1\}$ shows whether site j has a replica of site i or not. D_i denotes the used capacity, i.e., the amount of primary data at site i, which does not include the amount of replication data from other sites. P_{ij} denotes the risk indicator representing the probability of damage to both site i and site j, and n is the total number of sites. This definition of the objective function Eq. (1) denotes the total amount of data expected to be damaged for a combination of variables x_{ij} when the number of replicas of each site is one. Therefore, by minimizing the objective function Eq. (1), we can obtain the highest availability solution against target disasters.

2) Constraints

RDR uses two constraints: a redundancy constraint and a storage capacity constraint.

The redundancy constraint is used to regulate the number of replicas to be created by a primary site in a replication process. This constraint is necessary because without it every site tries to

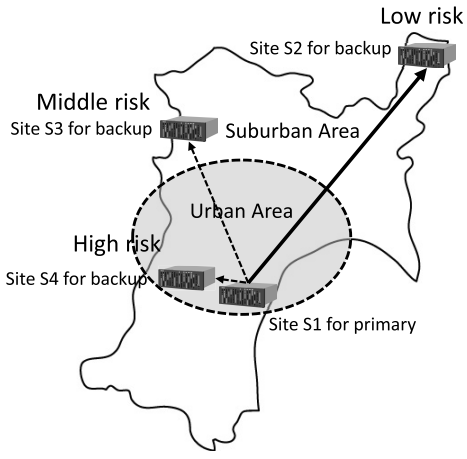


Fig. 1 Conceptual diagram of risk-aware data replication feature.

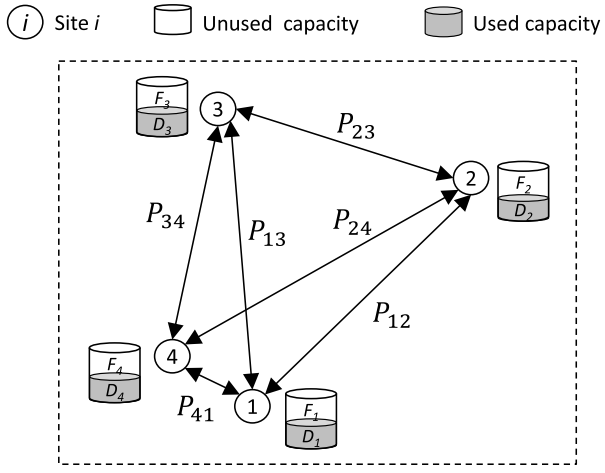


Fig. 2 Relation among sites in the case of 4 sites.

create as many replicas as possible. Consequently, the redundancy constraint is set to bound the maximum value of data redundancy. It is described as

$$\sum_{j=1}^{n} x_{ij} = R_i, \forall i, \tag{2}$$

where R_i, which denotes the number of replicas of site i, is given by the administrator of each site. We use $R_i = 1$ for the following discussion.

The storage capacity constraint is used to regulate the number of replicas that a backup site receives in a replication process. This constraint is necessary because every site has a finite storage capacity. Consequently, the storage capacity constraint is set to bound the maximum value of storage capacity consumed by replicas. It is described as

$$\sum_{i=1}^{n} D_i x_{ij} \leq F_j, \forall j, \tag{3}$$

where F_j denotes the unused capacity, i.e., the storage capacity of site j. It is given by the administrator of each site.

2.4 Issues in Applying Risk-aware Data Replication to an Urban and Suburban Area

If the backup site decided by IPP is located in the outer suburban area like Fig. 1, it becomes difficult for citizens living in the urban area to access the information desired by them because severe disasters can damage information networks as described in Section 1. Under such circumstances, they have to go to the far backup site physically to access the information.

Therefore, when the field to apply RDR is extended to the suburban area in addition to the urban area, it becomes difficult for them to access the information.

3. Distance Limiting Methods

In this section, we present overview of distance limiting methods and its usage cases. Then, we propose the average distance limiting (ADL) method and the maximum distance limiting (MDL) method to approach the issues described in the previous section.

3.1 Overview of Distance Limiting Methods

To address the issues, it is a feasible idea to limit the distance between primary site and backup site. There are two approaches: One is to limit the average distance, the other is to limit the maximum distance. The detail of each approach is described in the following subsections.

Figure 3 shows the usage cases of the distance limiting methods. Once a disaster damaging a primary site occurs, citizens living near the primary site can go to the backup site by walking, bicycle, or car. Moreover, staff who are responsible for an information service can deliver storage devices such as HDDs, flash media, or magnetic tapes at the backup site to the primary substitution site by walking, bicycle, car, or drone.

The appropriate value of distance limitation depends on the means of mobility or delivery.

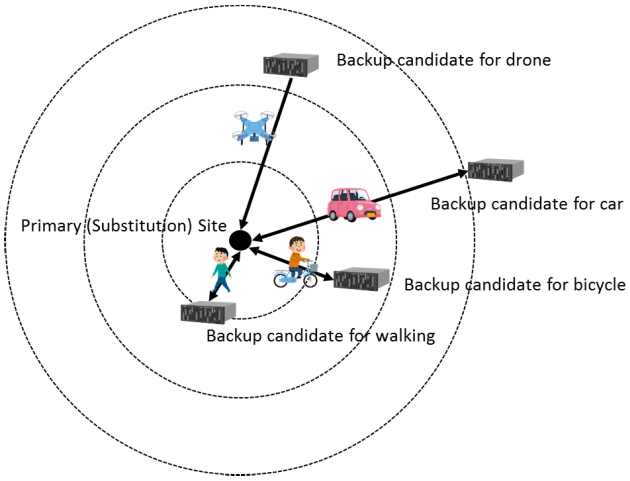


Fig. 3 Usage cases of distance limiting methods.

3.2 Average Distance Limiting Method

The ADL method limits the average distance of primary-backup site pairs. This can be expressed in an additional constraint of IPP. It is described as

$$\sum_{i=1, i\neq j}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \leq n \cdot d_a, \tag{4}$$

where d_{ij} denotes the moving distance between site i and site j, d_a denotes the upper limit of the average distance of primary-backup site pairs.

The ADL method is expected to give a solution with the average distance close to d_a. Therefore, this method will give a higher data availability. However this method may consume a lot of computation time because the constraint is related to all of the variables. The number of variables for this method (n_a) does not change from the original objective function Eq. (1) and $n_a = n(n-1)$. Then, the number of all combinations is 2^{n_a}. Therefore, the order of computation time is basically exponential. However, as the Branch-and-Bound method [7] described later in detail can cut a branch of a cluster of combinations without computation for each evaluation, the dominant of computation time often becomes Linear Programming Problem (LPP). Moreover, the computation order of double simplex algorithms for LPP is known as $O({n_a}^4)$. Although the class of this problem is nondeterministic polynomial (NP), the order of computation time is $O({n_a}^4)$ when some algorithms work well. As these discussions are also influenced by the symmetry and the distribution of the coefficient of the objective function, a generalized discussion is by no means easy.

From the perspective of a practice, it may be difficult to apply this method to the carrier having the hard limit of the cruising distance such as the drone.

3.3 Maximum Distance Limiting Method

The MDL method limits the maximum distance of primary-backup site pairs. There are two forms to express this method.

The first form is expressed in additional constraints of IPP. It is described as

$$d_{ij} x_{ij} \leq d_m, \forall i, j, \tag{5}$$

where d_m denotes the maximum distance of primary-backup site pairs.

The second form is expressed in the revised objective function. It is described as

$$f'(x_{12}, \cdots, x_{n(n-1)}) = \sum_{(i,j)\in S} D_i P_{ij} x_{ij}, \tag{6}$$

where S denotes the set of (i, j) elements which satisfy the equation $d_{ij} \leq d_m$.

The MDL method, especially expressed by the second form, is expected to give a solution in short computation time. Because the revised objective function Eq. (6) reduces the number of variables from the original objective function Eq. (1). In the same way in the previous subsection, the order of computation time of this method is $O((n_a - n_e)^4)$ where n_e denotes the number of eliminable variables. The value of n_e increases with a decrease in d_m. On the other hand, this method may give lower data availability than the ADL method when the same value is given to d_a and d_m.

4. Simulation Setup

We present details of simulation to be used for the evaluation in the next section.

4.1 Simulation Procedure

To evaluate data availability, we use an RDR simulator comprising four program modules as depicted in **Fig. 4**: a field creator, a risk calculator, a pair creator, and a disaster injector.

The field creator simulates sites in an area such as flat ground, a slope, a mountain, or the sea, and outputs field information and site information. Field information includes the geological formations of the ground at specified coordinates of the computational mesh. The site information includes the site locations, their usage, and the free storage capacity.

The risk calculator calculates the risk indicator P_{ij} for all pairs of sites with risk hints. The method of calculating the risk indicators is described in the next subsection.

The pair creator seeks combinations of safe pairs of primary-backup sites. It formulates and solves the IPP using constraint information and a specified algorithm. The constraint information includes data redundancy. The algorithm to solve IPP can be selected from the Branch-and-Bound (BB) method [7] or the Greedy method [8]. The BB method is a well-known general-purpose algorithm that is guaranteed to seek the optimal combination. The Greedy method is also a well-known general-purpose algorithm that does not guarantee an optimal combination. However, the computation time of the Greedy method is much shorter than that of the BB method. The pair creator invokes lp_solve 5.5, a well-known IPP solver command line interface, with no options, for the BB method. For the Greedy method the pair creator uses our own subroutines. The output is a list of the primary-backup site-pairs.

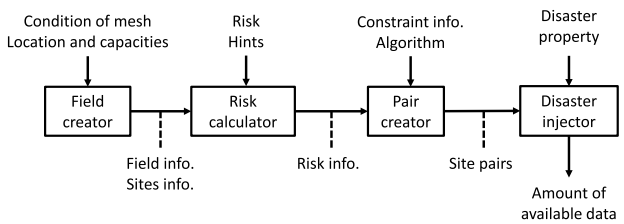


Fig. 4 RDR simulator.

The disaster injector generates a disaster and damages some sites according to the input disaster properties. Each site is determined to be either safe or damaged based on the damage probability described in the next subsection.

Finally, the amount of surviving unique data after the simulated disaster is calculated to compare data availability. It is calculated by subtracting the amount of lost unique data from the total amount of unique data. The amount of lost unique data is calculated by a summation of the amount of unique data at each primary site when both the primary site and its backup site are damaged by the simulated disaster.

4.2 Risk Indicator and Disaster Model

We use earthquakes in the simulation. Because earthquakes are one of general disasters and it is relatively easy to create a model with previous studies. This should be a good first step to apply to other types of disasters. In this subsection, we present a risk indicator and a disaster model for an earthquake.

4.2.1 Risk Indicator for an Earthquake

According to a physical model of earthauakes [9], the earthquake strength is calculated from the earthquake magnitude, the depth of its hypocenter, and the distance between its hypocenter and the site. However, nobody can predict the hypocenter or magnitude of a coming earthquke accurately. For this reason, it is difficult to use this information as a risk indicator for now. To overcome this difficulty, we consider the following two perspectives.

The first perspective is the applied physical model. Regardless of above, we know that the area damaged by an earthquake has geographical locality, which means that the risk indicator of a site i and another site j far from site i might be low. Therefore, we apply the site distance between the two sites instead of the distance between its hypocenter and the site to the original equation of the earthquake strength. According to the original equation, the earthquake strength decreases with an increasing distance in proportion to the negative value of logarithm of the distance.

The second perspective is the ease of use. It must be easy to use in the simulation if the risk indicator has characteristics of a probability. The probability of data loss should be positive and converge to 1 with an increasing earthquake strength. Therefore, it is a possible idea to use sigmoid function for the risk indicator.

Consequently, with consideration for above two perspectives, we describe risk indicator for an earthquake as

$$P_{ij}(d'_{ij}) = \zeta(a(-\log_{10} d'_{ij} - b)), \tag{7}$$

where d'_{ij} denotes the direct distance between site i and site j. Actually, a, b are design parameters that are used to tune the curve P_{ij}. These design parameters are calculated using linear interpolation with two pairs of (d'_{ij}, P_{ij}) as a risk hint in the input variable space of the sigmoid function ζ.

As this model is very basic, the differences of the ground strength and the building strength for each site are not considered. Therefore, this model is insufficient for covering the region

of abnormal seismic intencity. Further improvement of the accuracy of the risk indicator is a subject for future work. One idea is to apply the predicted probability value in the hazard map to the risk indicator.

4.2.2 Disaster Model for an Earthquake

The disaster model for an earthquake calculates the probability of damage at each site. It is used to decide whether each site will survive or be damaged based on stochastic simulations described in the following section. The damage probability Q for each site is

$$Q = \zeta(\alpha(T - \beta)), \tag{8}$$

where T denotes the earthquake strength at each site and is calculated from the earthquake magnitude, the depth of its hypocenter, and the distance between its hypocenter and the site. α is the "slope," a parameter to tune the increasing rate of Q, and β is designated as a "central value," a parameter to tune the strength T. Additional information is presented in Refs. [5], [6]. The physical model of the earthquake strength is also based on Ref. [9].

4.3 Simulation Conditions

The RDR simulator conditions are described in this subsection.

First, we set the input information for the field creator as shown next. The field size is 60 km × 60 km and the field height is 10 m. The sites are located randomly in the field. We set the number of sites in the field to several values from 20 to 2,000 shown in **Table 1**. This is because we assume the use of massively multiple and small-sized facilities such as pharmacies described in Section 2.2. We prepare 10 random patterns of sites location with respect to each of the number of sites. Each site has one datum to back up, and is capable of receiving one datum for backing up other sites. This setting is equivalent to each site having one storage volume to back up, and having one storage volume to store the backup data of other sites. Note that the replication sites are allocated not in this phase but in the pair creator phase. Therefore, the replication sites themselves are not randomly allocated. To apply the proposed methods to large-sized facilities such as data centers, we should set more complex conditions such that each site does not have equal but varying amounts of used and unused capacity. Then, the relations of replication sites of such facilities may be like scale-free network. This remains as a subject for future work.

Secondly, we set the input information for the risk calculator as shown next. We set the risk hints $(d'_{ij}, P_{ij}) = (5, 0.2), (20, 0.1)$ to determine the design parameters a, b for an earthquake. This represents a situation in which a strong earthquake damages 20% of the sites at a distance of 5 km and 10% of the sites at a distance of 20 km. The combination of these values is selected from four combination candidates of risk hints because it fitted well to the result of an injected virtual disaster.

Table 1 Simulation parameters.

Parameters	Values
Number of sites	20, 40, 80, 100, 200 (for all methods) 400, 800, 1000, 2000 (except for combination of ADL method and BB method)
Limiting distance (km)	5, 10, 15, 20, 25, 30

Thirdly, we set the input information for the pair creator as shown next. We use the two algorithms described in Section 4.1: the BB method and the Greedy method. We set the number of replicas for each site to one. We set the limiting distance as the value shown in Table 1. We use the same value of the limiting distance to d_a and d_m in this paper.

Fourth, we set the input information for the disaster injector as shown next. We set a magnitude eight earthquake with a different hypocenter. The X-Y location of hypocenter is randomly decided in the field and the depth (Z) of hypocenter is fixed to 50 km. The simulation is executed 500 times while changing the X-Y location of hypocenter. We adjust and set parameters such as α, β for the earthquake to damage about 40% of the total number of sites on average.

We use a server with a processor (Intel Xeon E5502, 1.87 GHz, 2 cores, 1 chip) and 6 GB RAM for RDR simulation.

5. Evaluation

In this section, we evaluate the proposed method from the perspectives of the distance distribution of sites, the data availability, and the computation time for site pairing.

In the evaluation, we compare four combinations of methods: (1) the ADL method and the BB method, (2) the MDL method and the BB method, (3) the ADL method and the Greedy method, (4) the MDL method and the Greedy method. We call the four combinations in the following as the ADL-BB method, the MDL-BB method, the ADL-G method, the MDL-G method, respectively.

5.1 Distance Distribution of Sites

As described in Section 4, primary-backup site pairs are generated by the pair creator when the number of sites, the sites location, and the limiting distance are fixed. Then, the site pairs are commonly used in 500 times simulation of the disaster injector. **Figure 5** shows the distance distribution of sites based on the site pairs with the conditions that the number of site is 200 and the limiting distance is 20 km. The vertical axis is log scale and presents the percentage of sites in the range. We discuss the result with the above conditions as the other results have similar characteristics. The ADL-BB method has a narrow distribution around 20 km. The MDL-BB method also has a narrow distribution less than 20 km. These two methods are expected to achieve the high data availability because almost all distance of primary-backup site pairs are close to 20 km which is the limiting distance.

The MDL-G method has a similar shape with the MDL-BB method except that it has a long tail to small distance. As reported in the previous papers [5], [6], the Greedy method sometimes gives inappropriate site pairs. In the same manner as the Greedy method, the MDL-G method tries to select site pairs with a long distance as much as possible. However, as the maximum distance is limited, it does not consume good backup site candidates for the next primary site unlike the Greedy method. For this reason, the MDL method may suppress the weak point of the Greedy method especially when the maximum distance is short. Therefore, the MDL-G method is also expected to achieve high

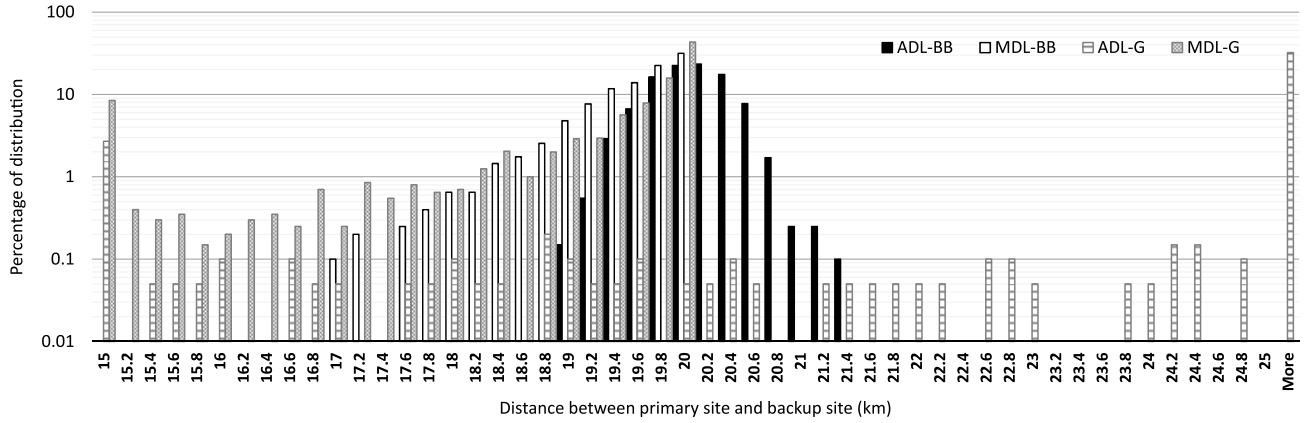


Fig. 5 Distance distribution of sites (200 sites, distance limitation: 20 km).

data availability in some cases.

On the other hand, the ADL-G method has a very wide distribution. Moreover, the ADL-G method could not match the backup sites for most of the primary sites because the Greedy method selects primary-backup site pairs with a very long distance in the early phase and the selection makes the average of the distance quite large. Therefore the ADL-G method is not expected to achieve high data availability.

5.2 Data Availability

Figure 6 shows the average data availability of best cases as a function of the number of sites. Each marker presents the average available data ratio of the method achieving the best in four methods. From 20 sites to 200 sites, the ADL-BB method achieves the best average data availability. From 400 sites to 2,000 sites, the MDL-BB method achieves the best of the average data availability. The average data availability becomes lower as a decreasing distance limitation.

From the perspective of the data availability, the ADL-BB method must be the most preferable method. Unfortunately, we have no result of the ADL-BB method from 400 sites to 2,000 sites because of long computation time for site pairing. However it is expected that there is no great difference of the data availability between the ADL-BB method from 20 sites to 200 sites and the ADL-BB method from 400 sites to 2,000 sites as the data availability stays almost unchanged against the number of sites.

Figure 7 shows the relative data availability from the data availability of the best method shown in Fig. 6 with the limiting distance of 5 km and 30 km. The relative results from 10 km to 25 km are not shown in this paper to save pages as those have similar characteristics. The MDL-BB method and the MDL-G method approach the ADL-BB method as the number of sites increases. When the number of sites is 200, the difference of the data availability between the ADL-BB method and the MDL-BB method is smaller than 0.5 point. Additionally, the difference of the data availability between the ADL-BB method and the MDL-G method is around 1 point. Therefore, the MDL-BB method and the MDL-G method are acceptable methods when the number of sites is greater than 100 in this condition. By contrast, there is no good point in the ADL-G method from the perspective of the data availability.

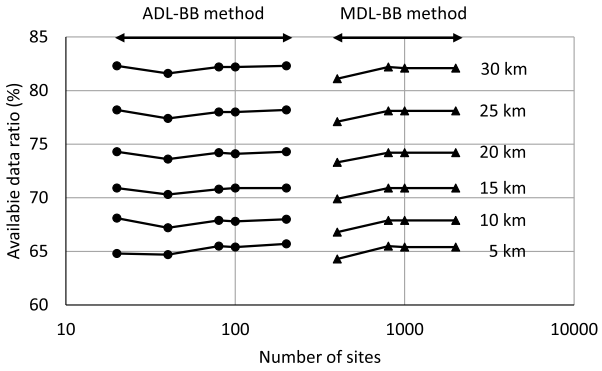


Fig. 6 Average data availability of the method achieving the best in four methods.

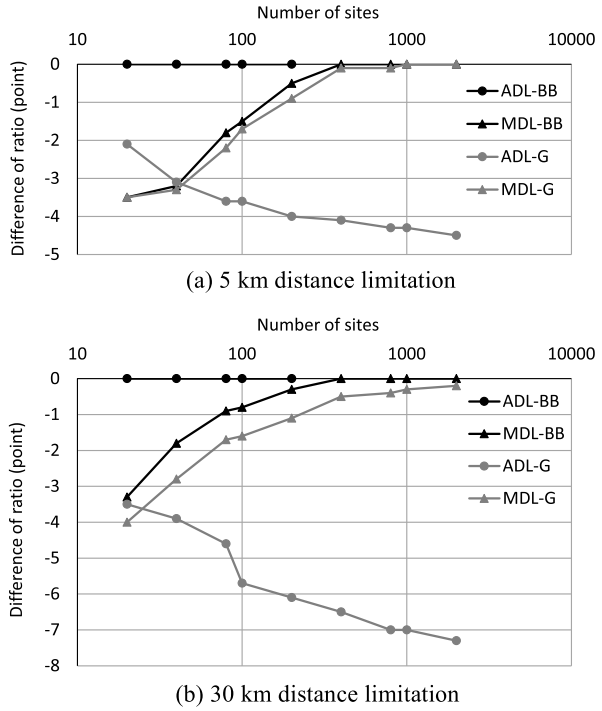


Fig. 7 Relative data availability from the best method.

As to MDL-G method, the difference of the data availability between the MDL-G method and the MDL-BB method which gives an optimal solution is smaller than 0.5 point under 5 km distance limitation. Under 30 km distance limitation, the difference is around 1 point. As described in the previous subsection, these

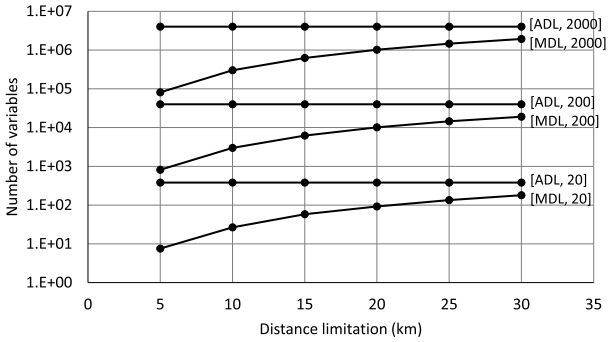


Fig. 8 Average number of variables of objective function.

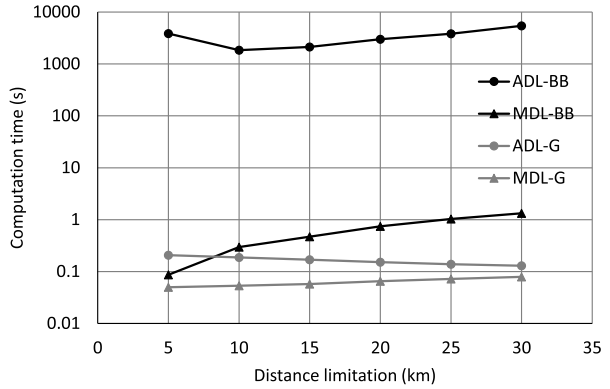
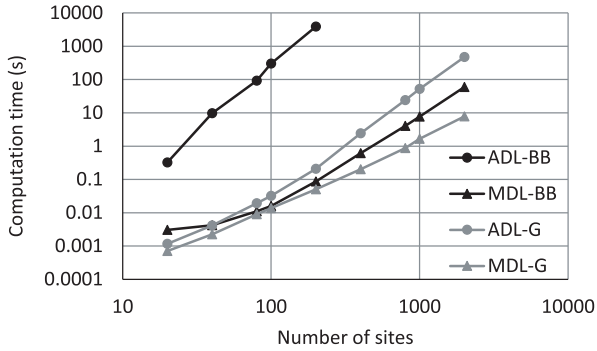
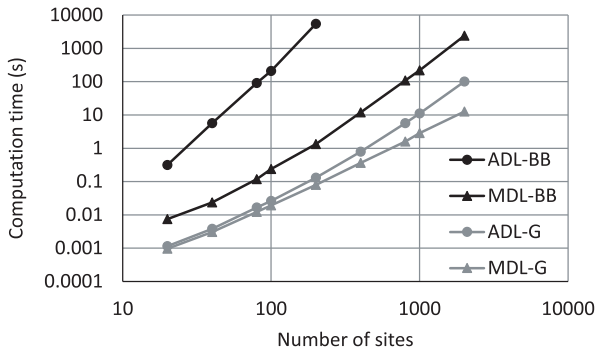


Fig. 9 Computation time of 200 sites pairing for each method as a function of distance limitation.



(a) 5 km distance limitation



(b) 30 km distance limitation

Fig. 10 Computation time of sites pairing for each method as a function of the number of sites.

results show the MDL method suppresses well the weak point of the Greedy method especially when the maximum distance is short.

5.3 Computation Time for Site Pairing

Figure 8 shows the average number of variables of objective function as a function of the distance limitation. The vertical axis is log scale. Each line has [method, num] symbol. The "method" shows the type of distance limiting methods and the "num" shows the number of sites. For example, [MDL, 20] means the MDL method and 20 sites.

The number of variables of the MDL method is smaller than the ADL method and becomes smaller as a decreasing distance limitation. In contrast, the number of variables of the ADL method does not change against the value of distance limitation. Therefore, it is expected that computation time of the MDL method is smaller than the ADL method especially for the short distance limitation.

Figure 9 shows the computation time of 200 sites pairing for each method as a function of distance limitation. The vertical axis is log scale. The computation time of the MDL-BB method, the ADL-G method, and the MDL-G method are smaller than the ADL-BB method by over 3 orders of magnitude. The computation time of the MDL-BB method and the MDL-G method slowly grows with the increasing distance limitation. This is because the number of variables increases with a greater distance limitation as shown in Fig. 8. The computation time of the ADL-G method slowly increases with the decreasing distance limitation. This is because the ADL-G method becomes hard to find backup sites to stay within the constraints with the decreasing distance limitation.

Figure 10 shows the computation time of site paring for each method as a function of the number of sites when the limiting distance is 5 km and 30 km. The results from 10 km to 25 km are not shown in this paper due to the same reasons. Both axes are log scale. The computation time of the ADL-BB method rapidly grows with the increasing number of sites. As the computation time is beyond a few hour or a week in some cases, it is hard to apply the ADL-BB method to more than a few hundreds sites. The computation time of the other methods also grows with the increasing number of sites although those are not so rapid. It is possible to apply those methods to less than a few thousand sites.

From the comprehensive perspectives, the MDL-BB method is the most practicable especially for more than a few hundreds sites. The MDL-G method is also practicable if a small degradation of the data availability is acceptable. The ADL-BB method is practicable only for less than a few hundreds sites.

6. Related Work

This section presents related work in terms of highly available information systems and fast algorithms to solve IPP. First, related work on highly available information systems is described.

Dynamo [10] distributes data over a set of nodes (i.e., storage hosts) based on a consistent hashing technique [11]. The Dynamo system determines replication targets using a "ring" created by the output range of a circular hash function. Each node is assigned a random value within the ring space, and replicates data to its successors. Dynamo distributes data fundamentally in a random manner because it partitions the hash range randomly when adding a new node to the system. It ignores the node's safety

against widespread disasters. Therefore, Dynamo requires high data redundancy if we hope for high availability in times of such disasters.

Cassandra [12] also distributes data over a set of nodes based on a consistent hashing technique. Additionally, it provides various replication policies such as "Rack Aware" and "Datacenter Aware". By activating these policies, the Cassandra system replicates data to different racks or different datacenters from the node storing the primary data. These policies enable the system to avoid failures related to a power outage, cooling failures, network failures, or natural disasters. Cassandra considers only whether the replication target is installed in the same rack or the same datacenter as the primary node, and never considers the node's safety against a widespread disaster. Therefore, Cassandra also requires high data redundancy if one hopes for high availability in times of such disasters.

Additionally, some distributed file systems such as Gluster FS [13], Ceph [14], and XtreemFS [15] have geo-replication features. These features are mainly designed for a small number of data centers and backup sites that are distant from primary sites. XtreemFS can select several replication policies including data center grouping. The data center grouping policy chooses multiple data centers that are closest to the client for storing replicas without consideration of the disaster risk.

Some research works have produced a disaster-resilient information system [16], [17]. Such works specifically examine the content placement of primary data, rather than backup data. Numerical evaluation specifically examines attacks by weapons of mass destruction (WMD). In their evaluation, the number of sites is small. Therefore, no proposal of original fast algorithms exists to solve IPP in the papers.

Secondly, related work of fast algorithms to solve IPP is described. No heuristic algorithm has been proposed for the IPP form in this paper, although the form is similar to the existing Multiple Knapsack Problem [18]. Therefore, three general-purpose algorithms for IPP are presented: the Branch-and-Bound method (BB method) [7], the Dynamic Programming method (DP method) [19], and the Divide and Conquer (D&C) algorithm [20].

The BB method is fundamentally an enumeration method, but it avoids searching useless branches without solution optimality by limiting the search range using an upper bound and lower bound of the solution of the relaxation problem.

The DP method creates partial problems from an original problem, then solves and temporarily stores the solutions of the partial problems, and finally solves the original problem using the stored solutions. As a conceptual method, it must be specialized to each target problem. A specialized algorithm is the Bellman–Ford method [21], which solves the shortest path problem, another form of IPP.

The D&C algorithm also breaks a problem into sub-problems. It is recognized as a key component of the DP method. The difference from the DP method is that all sub-problems are independent. Therefore D&C algorithm doesn't reuse the result of sub-problems. It is possible to apply the D&C algorithm to the problem in this paper, although it loses optimality. Moreover, dividing the area into the sub-area in response to each distance limitation is a complex task.

7. Conclusions

We propose distance limiting methods to be able to access urgently required data in the backup site from the damaged area around the primary site by physically transfer for risk-aware data replication (RDR). We present two possible methods: the average distance limiting (ADL) method and the maximum distance limiting (MDL) method. The ADL method limits the average distance of primary-backup site pairs. The MDL method limits the maximum distance of primary-backup site pairs. We evaluated its distance distribution, data availability and computation time by using RDR simulator. The results show that the combination of the MDL method and the Branch-and-Bound (MDL-BB) method is the most practicable from the perspectives of the data availability and the computation time especially for more than a few hundreds sites. In addition, the combination of the ADL method and the BB (ADL-BB) method is also practicable only for less than a few hundreds sites. We conclude that both methods can be used as the requirements such as the number of sites and the desired data availability.

Acknowledgments The authors thank Masatoshi Shimbori and Hiroshi Ichinomiya for their research computing support and all members of our project team for their useful comments. This work is supported as "Research and Development on Highly Functional and Highly Available Information Storage Technology," sponsored by the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] Patterson, R.H., Manley, S., Federwisch, M., Hitz, D., Kleiman, S. and Owara, S.: SnapMirror: File-system-based asynchronous mirroring for disaster recovery, *Proc. 1st USENIX Conference on File and Storage Technologies* (*FAST*), pp.117–129 (2002).

[2] EMC Education Services: Remote Replication, *Information Storage and Management*, *2nd ed*., pp.289–310, John Wiley & Sons, Inc. (2012).

[3] Hitachi Data Systems: Disaster Recovery Issues and Solutions (a white paper), available from ⟨http://www.hds.com/assets/pdf/wp_117_02_disaster_recovery.pdf⟩.

[4] Kobayashi, M.: Experience of Infrastructure Damage Caused by the Great East Japan Earthquake and Countermeasures against Future Disasters, *IEEE Communications Magazine*, Vol.52, No.3, pp.23–29 (2014).

[5] Matsumoto, S., Nakamura, T. and Muraoka, H.: Risk-aware Data Replication to Massively Multi-sites against Widespread Disasters, *Proc. 2nd Asian Conference on Information Systems* (*ACIS*), p.34 (2013).

[6] Matsumoto, S., Nakamura, T. and Muraoka, H.: Risk-aware Data Replication to Massively Multi-sites against Widespread Disasters, *Rangsit Journal of Information Technology*, Vol.1, No.2, pp.22–28 (2013).

[7] Lawler, E.L. and Wood, D.E.: Branch-and-Bound Methods: A survey, *Operations Research July/August*, Vol.14, No.4, pp.699–719 (1996).

[8] Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem, *Proc. American Mathematical Society*, Vol.7, No.1, pp.48–50 (1956).

[9] Si, H. and Midorikawa, S.: New attenuation relations for peak ground acceleration and velocity considering effects of fault type and site condition, *Proc. 12th World Conference on Earthquake Engineering* (*WCEE*), CD-ROM, No.532 (2000).

[10] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Pilchin, A., Sivasubramanian, S., Vosshall, P. and Vogels, W.: Dynamo: Amazon's highly available key-value Store, *Proc. 21st ACM SIGOPS Symposium on Operating Systems Principles* (*SOSP*), pp.205–220 (2007).

[11] Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M. and Lewin, D.: Consistent hashing and random trees: Distributed caching

protocols for relieving hot spots on the World Wide Web, *Proc. 29th Annual ACM Symposium on Theory of Computing* (*STOC*), pp.654–663 (1997).
[12] Lakshman, A. and Malik, P.: Cassandra: A de-centralized structured storage system, *ACM SIGOPS Operating Systems Review*, Vol.44, No.2, pp.35–40 (2010).
[13] Muntimadugu, D.: Gluster File System 3.3.0 Administration Guide Using Gluster File System, available from ⟨http://www.gluster.org/wp-content/uploads/2012/05/Gluster_File_System-3.3.0-Administration_Guide-en-US.pdf⟩.
[14] Weil, S.A., Brandt, S.A., Miller, E.L., Long, D.D.E. and Maltzahn, C.: Ceph: A scalable, high-performance distributed file system, *Proc. 7th symposium on Operating systems design and implementation* (*OSDI*), pp.307–320 (2006).
[15] The XtreemFS Installation and User Guide Version 1.5.x, available from ⟨http://www.xtreemfs.org/xtfs-guide-1.5.pdf⟩.
[16] Ferdousi, S., Dikbiyik, F., Habib, M.F. and Mukherjee, B.: Disaster-Aware Data-Center and Content Placement in Cloud Networks, *Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems* (*ANTS*), pp.1–3 (2013).
[17] Savas, S.S., Dikbiyik, F., Habib, M.F. and Mukherjee, B.: Disaster-Aware Service Provisioning by Exploiting Multipath Routing with Manycasting in Telecom Networks, *Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems* (*ANTS*), pp.1–3 (2013).
[18] Martello, S. and Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, Inc. (1990).
[19] Bellman, R.: The Theory of Dynamic Programming, *Bulletin of the American Mathematical Society*, Vol.60, pp.503–515 (1954).
[20] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C.: *Introduction to Algorithms, 3rd ed.*, The MIT Press (2009).
[21] Bellman, R.: On a routing problem, *Quarterly of Applied Mathematics*, Vol.16, pp.87–90 (1958).

Takaki Nakamura received B.E, M.E, and Ph.D. in information science and technology from Osaka University in 1996, 1998, and 2011, respectively. He joined Central Research Laboratory, Hitachi, Ltd. in 1998. He has been an associate professor at Research Institute of Electrical Communication, Tohoku University since 2012. His research interests include storage systems, file systems, and operating systems.

Shinya Matsumoto received B.E. (2004) and M.E. (2006) degrees from Nagoya University, Japan. He currently works in the Storage Research Department, Center for Technology Innovation-Information and Telecommunications, Research & Development Group, Hitachi, Ltd., Japan. His current research interests include operating systems, file systems, and data management in storage systems.

Masaru Tezuka is the Manager of Research and Development Department at Hitachi Solutions East Japan, Ltd. He received B.E. and M.E. in bio-physical engineering from Osaka University, Japan and Ph.D. in systems and information engineering from Hokkaido University, Japan. His research interests include nonlinear optimization, evolutionary computation, computational intelligence, computational statistics, risk analysis, and their industrial application.

Satoru Izumi received M.S. and Ph.D. degrees from Tohoku University, Japan, in 2009 and 2012, respectively. He is currently a research fellow of Cyberscience Center at Tohoku University. His research interests include Semantic Web, green ICT and Disaster-resistant communications. He is a member of IEICE and IPSJ.

Hiroaki Muraoka received B.E, M.E, and Ph.D. in electrical engineering from Tohoku University in 1976, 1978, and 1981, respectively. He joined Matsushita Communication Industrial in 1981. He is currently a professor at Research Institute of Electrical Communication, Tohoku University. He has been engaged in research on high density perpendicular magnetic recording, disk drive technologies, and information storage systems.