Discreet Method to Match Safe Site-Pairs in Short Computation Time for Risk-Aware Data Replication

Takaki NAKAMURA†(a), Shinya MATSUMOTO††, Nonmembers, and Hiroaki MURAOKA†, Member

SUMMARY Risk-aware Data Replication (RDR), which replicates data at primary sites to nearby safe backup sites, has been proposed to mitigate 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 damage risk for both the primary site and the backup candidate site. To minimize the damage risk of all site-pairs the Integer Programming Problem (IPP), which is a mathematical optimization problem, is applied. A challenge for RDR is to choose safe backup sites within a short computation time even for a huge number of sites. As described in this paper, we propose a Discreet method for RDR to surmount this hurdle. The Discreet method first judges the backup sites of a potentially unsafe primary site and avoids assigning a very safe primary site with a very safe backup site. We evaluated the computation time for site-pairing and the data availability in the cases of Earthquake and Tsunami using basic disaster simulations. We confirmed that the computation rate of the proposed method is more than 1000 times faster than the existing method when the number of sites is greater than 1000. We also confirmed the data availability of the proposed method; it provides almost equal rates to existing methods of strict optimization. These results mean that the proposed method makes RDR more practical for massively multiple sites.

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

1. Introduction

Remote replication [1]–[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, extending outside of the disaster area as well as information servers in the disaster area. Therefore, people in the disaster area become isolated from the disaster site area via the wide area network including the internet. Under such severe circumstances, the continuation of information services is limited by existing technologies to areas outside of 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 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 transporting the surviving storage apparatus on a network-isolated site. In previous studies [5], [6], formulation of the Integer Programming Problem for RDR was shown. Improved results of data availability using the Branch-and-Bound (BB) method [7] and the Greedy method [8] were confirmed. However, it remains a challenge to improve data availability sufficiently within a short computation time for disasters of multiple types.

In this paper, we propose the Discreet method, a heuristic algorithm, to overcome this challenge. The proposed method first selects the backup sites of potentially unsafe primary sites and avoids assigning a very safe primary site with a very safe backup site.

The remainder of this paper is organized as follows. We explain RDR and the remaining challenges in Sect. 2. Then we propose the Discreet method, a heuristic algorithm, to decide primary-backup site-pairs of RDR in Sect. 3. In Sect. 4, we describe the risk indicator and disaster model used in the simulation. In Sect. 5, we compare the proposed method and the existing algorithms in terms of data availability and computation time for site-pairing. We present related work and conclusions in Sects. 6 and 7, respectively.

2. Risk-Aware Data Replication Features and Challenges

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

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

age risks for both the primary site and the backup site. Figure 1 shows a conceptual diagram of RDR. The east is the coastal side and the west is the inland side. Four sites exist in the area, each with an information server. The information server at site S1 operates an information service. The three candidate backup sites are sites S2, S3, and S4. We assume that this area will be damaged by a tsunami in the near future. Because the lowest risk site for a tsunami 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 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 Programming 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

$$f(x_{12}, \ldots, x_{n(n-1)}) = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} P_{ij} x_{ij},$$  \hspace{1cm} (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 $j$. $n$ is the total number of sites. This definition of the objective function (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 (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 create as many replicas as possible. Consequently, the redundancy constraint is set to bound the maximum value of data redundancy. It is described as

$$\sum_{j=1}^{n} x_{ij} = R_i, \forall i,$$  \hspace{1cm} (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

$$\sum_{i=1}^{n} D_{ij} x_{ij} \leq F_j, \forall j,$$  \hspace{1cm} (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.

2.3 Challenges of Risk-Aware Data Replication

In earlier studies [5], [6], two existing algorithms were applied to the form of IPP in the previous subsection. One is the Branch-and-Bound (BB) method [7]: a well-known general-purpose algorithm that is guaranteed to seek the optimal combination. The other is the Greedy method [8], which is also a well-known general-purpose algorithm and which does not guarantee an optimal combination. However, the computation time of the Greedy method is much shorter than that of the BB method.
The results of the previous study presented two issues. First, the combination pairs of primary-backup sites generated by the Greedy method sometimes failed to improve data availability under particular types of disasters such as near-field earthquakes. Secondly, generation of combination pairs using the BB method was extremely time consuming when the number of sites extended beyond the hundreds. The long computation time complicates system installation and changing of the system configuration.

Therefore, the challenge of RDR is to improve data availability sufficiently for disasters of multiple types within a short computation time, even when the number of sites extends beyond the hundreds.

3. Discreet Method

In this section, we propose a method to overcome the challenge presented in the previous section.

3.1 Overview of Discreet Method

The two basic design policies of the proposed method are as follows:
(1) Avoid pairs of primary-backup sites that provide too much safety
(2) Limit computing operations to \((n^3)\)

We call the proposed method “Discreet” because it chooses the site pairs not greedily but discreetly based on the basic design policy (1). The Discreet method has two phases: nomination and trade. In the nomination phase, a backup site candidate for each primary site is nominated. In the trade phase, the backup site candidates for two primary sites are swapped if the value of the objective function is improved. The following subsections present details of the two phases.

3.2 Phase 1: Nomination Phase

In the nomination phase, backup site candidates are nominated. The backup site is not fixed in this phase. To simplify the explanation, we use a condition by which the values of \(D_i\) and \(F_j\) are set to 1 in the remainder of this section. The steps of the nomination phase under these conditions are as follows:

(1) Select a backup site with the lowest value of \(P_{ij}\) from all backup sites for each primary site \(i\). The site number of the backup site with the lowest value for primary site \(i\) is defined as \(j_{min}(i)\). The selection is repeated for every primary site \(i\).

(2) Select a primary site with the highest value of \(P_{i,j_{min}(i)}\) from all primary sites. The site number of the primary site with the highest value is defined as \(i_{max}(k)\) (initial value of \(k = 1\)).

(3) Define the site number of the backup site candidate for primary site \(i_{max}(k)\) as \(j_{min}(i_{max}(k))\).

(4) Repeat the next round from (1) with an increment of \(k\) after removing primary site \(i_{max}(k)\) and backup site \(j_{min}(i_{max}(k))\) from the list of candidate sites unless no primary site has a backup site candidate.

Figure 3 presents an example of four iterations of the nomination phase. Each row corresponds to a primary site. Each column corresponds to a backup site. Each element of the matrix in the \(i\)-th row and \(j\)-th column corresponds to the risk indicator \(P_{ij}\). The first matrix shows the state after the first round \((k = 1)\). The first backup site candidate is chosen using this matrix. The right list shows the lowest value of the risk indicator for each primary site. Each element of the matrix in the \(i\)-th row and \(j\)-th column corresponds to the risk indicator \(P_{ij}\).

The first matrix shows the state after the first round \((k = 1)\). The first backup site candidate is chosen using this matrix. The right list shows the lowest value of the risk indicator for each primary site extracted from the first matrix. The values for primary sites 1-4 are respectively, 0.1, 0.2, 0.3, and 0.6. Because the highest value in the list is 0.6, site 1 is selected as the backup site candidate for primary site 4.

The second matrix from the top shows the state after the second round \((k = 2)\). The second backup site candidate is selected according to this matrix. Elements in black are excluded from selection for two reasons. The first reason is that backup sites 2 and 3 are no longer backup site candidates for primary site 4. The second reason is that the free space at backup site 1 has been allocated to primary site 4. After following the same routine as that used in the first round, site 3 is selected as the backup site candidate for primary site 4.

The second matrix from the top shows the state after the second round \((k = 2)\). The second backup site candidate is selected according to this matrix. Elements in black are excluded from selection for two reasons. The first reason is that backup sites 2 and 3 are no longer backup site candidates for primary site 4. The second reason is that the free space at backup site 1 has been allocated to primary site 4. After following the same routine as that used in the first round, site 3 is selected as the backup site candidate for primary site 4.

The third and fourth candidates are decided in the same way. Finally, the backup site candidates for primary sites 1-4 are decided as 4, 3, 2, and 1, respectively.

3.3 Phase 2: Trade Phase

In the trade phase, swapping the two backup site candidates
of two primary sites is tried. In the nomination phase, the backup site candidates are not always low-risk sites when \( k \) is large. The purpose of the trade phase is to improve such candidates. The swaps are executed as follows:

1. Select the backup site candidate \( j \) of a primary site \( i \) with the highest value of \( P_{ij} \) among all the backup site candidates decided in the nomination phase. The site numbers of the backup site candidate selected in this step and the primary site are defined, respectively, as \( j^*(k) \) and \( i^*(k) \) (initial value of \( k = 1 \)).

2. Select another backup site candidate \( j''(k) \) of primary site \( i''(k) \) which most improves the risk indicator \( P_{ij} \) when the backup site candidates \( j(k) \) and \( j''(k) \) are swapped. Exit this phase if no such backup site candidate exists.

3. Swap the candidate backup sites. The backup site candidate for primary site \( i^*(k) \) becomes \( j''(k) \). The backup site candidate for primary site \( i''(k) \) becomes \( j(k) \).

4. Repeat from (1) with an increment of \( k \).

Figure 4 presents an example of the steps in the trade phase. As shown in the previous subsection, the current backup site candidates for primary sites 1-4 are, respectively, 4, 3, 2, and 1. The first matrix shows the initial state. The highest value is 0.9 for the pairing of primary site 1 and backup site candidate 1. The second matrix shows the current state after the first swap with primary site 2. The highest value increases to 0.7. The third matrix shows the state after the second swap with primary site 3. The highest value remains at 0.7. Therefore, this is a possible trade.

Secondly, we try to swap a backup site candidate for primary site 3 and the target. The backup site candidate for primary site 1 changes to 2 from 4; the backup site candidate for primary site 3 changes to 4 from 2. After swapping the highest value in the right list becomes 0.6. This second trial yields a higher value of 0.7 than the first trial. Because no more primary sites exist for swapping, we make the trade of the second trial.

In this example, it is not possible to reduce the highest value by making further swaps and the backup site candidates are fixed. Therefore, the backup sites for primary sites 1-4 are finally decided as 2, 3, 4, and 1, respectively.

4. Examples of the Risk Indicator and Disaster Model

In this section, we discuss the risk indicators and disaster models for earthquakes and tsunami.

As described in Sect. 2.2, the risk indicator \( P_{ij} \) denotes the probability that both site \( i \) and site \( j \) are damaged. A system administrator can set the risk indicator flexibly.

A disaster model is necessary to evaluate the availability improvement by RDR. The disaster model shows damaged sites based on the disaster properties obtained from the input information.

In the following subsection, examples of the risk indicator and disaster model are shown for disasters of each type. These are used for evaluation in the following section.

4.1 Representation of an Earthquake

This subsection describes the risk indicator and disaster model for an earthquake.

4.1.1 Risk Indicator for an Earthquake

First, we discuss the risk indicator for an earthquake. Nobody can predict the hypocenter or magnitude of a coming earthquake accurately. Therefore, it is difficult to use this information as a risk indicator for now. Nevertheless, we know that the area damaged by an earthquake has geographical locality, which means that the risk indicator of a site and another site far from site \( i \) might be low. We use this characteristic. The risk indicator for an earthquake is described as

\[
P_{ij}(d_{ij}) = \zeta(a - \log_{10}d_{ij} - b),
\]

where \( d_{ij} \) denotes the 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 \( \zeta \).

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.

4.1.2 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

$$Q = \zeta(\alpha(T - \beta)), \quad (5)$$

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. $\alpha$ is the "slope," a parameter to tune the increasing rate of $Q$, and $\beta$ is designated as a "central value", a parameter to tune the strength $T$. Additional information is presented in [5], [6]. The physical model is based on [9].

4.2 Representation of a Tsunami

In this subsection, the risk indicator and disaster model for a tsunami are described.

4.2.1 Risk Indicator for a Tsunami

The tsunami risk indicator is calculated as

$$P_i(S_i) = \zeta(a(S_i - b)), \quad (6)$$

where $S_i$ denotes the tsunami strength and $a, b$ are design parameters used to tune the curve $P_i$. As described herein, we use the inundation height when the tsunami almost reaches the site with the greatest distance from the seacoast as $S_i$. The strength of the tsunami $S_i$ is

$$S_i = \frac{qZ_i}{X_i^2} \sqrt{\frac{X_i}{Z_i}}, \quad (7)$$

where $q$ denotes the cross-sectional area of the tsunami which overflows a breakwater. $X_i, Z_i$ respectively denote the relative $x$ and $z$ positions of site $i$ from the coast. We assume that the risk indicator has the characteristics of probability such that the risk indicator $P_{ij}$ is expressed as the product of $P_i$ and $P_j$.

4.2.2 Disaster Model for a Tsunami

The disaster model decides which sites become tsunami-damaged. We use a cumulative probability function for a fragility function [10] and damage probability $Q$ for each site described as

$$Q = \phi \left( \frac{x - \mu}{\sigma} \right), \quad (8)$$

where $x$ is the inundation depth, and $\mu$ and $\sigma$ respectively denote the mean and the standard deviation of $x$. We use parameters obtained through regression analysis of the 2004 Sumatra–Andaman earthquake tsunami for $\mu$ and $\sigma$. The inundation depth $x$ at each site is calculated using the level fill method [11]. In the level fill method, the tsunami height at the seacoast is used to calculate the inundation depth $x$. The tsunami height is calculated using the estimation model from magnitudes [12].

5. Evaluation

In this section, we discuss evaluation of the data availability and computation time.

5.1 Simulation Procedure

To evaluate data availability, we use an RDR simulator comprising four program modules as depicted in Fig. 5: 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 Sect. 4.

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 can be selected from the BB method, the Greedy method, or the Discreet method we propose in this paper. 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 and Discreet methods the pair creator uses our own subroutines. The output is a list of the primary-backup site-pairs.

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 Sect. 4.

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.

5.2 Simulation Conditions

The RDR simulator conditions are described in this subsec-
First, the input information for the field creator is depicted in Fig. 6. The left of the figure is the inland side and the right is the coastal side. The target system comprises 135 sites, which are located randomly in a field of 30 km × 20 km in size, excluding the sea area. The field height in the area from (0, 0)-(29, 20) (km) is 10 m. The height decreases linearly when \( x \) is greater than 29 km and the coastline is (30, 0)-(30, 20). 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.

Secondly, the input information for the risk calculator is as follows: For an earthquake, we set the risk hints \((d_{ij}, P_{ij}) = (5, 0.2), (20, 0.1)\) to determine the design parameters \(a, b\). 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. For a tsunami disaster, we set the risk hints \((d_i, z_i, P_i) = (0.5, 0.01, 0.9), (5, 0.03, 0.2)\) to determine design parameters \(a, b\). This represents a situation in which a strong tsunami damages 90% of the sites located 0.5 km from the coast at a height of 10 m, and 20% of the sites located 5 km from the coast at a height of 30 m.

Thirdly, the input information for the pair creator is as follows: We use the three algorithms described in the previous subsection: the Greedy method, the BB method, and the Discreet method. We set the number of replicas for each site to one.

Fourth, the input information for the disaster injector is as follows: We evaluate four types of disasters: (1) a near-field earthquake, (2) an earthquake under the coast line, (3) a far-field earthquake, and (4) a tsunami generated by an earthquake. Figure 6 and Table 1 show the hypocenter of each earthquake and the magnitude. We adjust and set parameters such as \(\alpha, \beta\) for the earthquake and the wave period for the tsunami to damage half of the total number of sites.

We use a server with a processor (2-core 1.87 GHz, Intel Xeon® E5502) and 6 GB RAM for RDR simulation. The simulation for a disaster of one type is executed 500 times. The simulator also invokes a Random method in addition to the three algorithms described in Sect. 5.1. The Random method chooses the backup sites in a random manner.

### 5.3 Simulation Results

In this subsection, we discuss the simulation results from the perspectives of the distance distribution of sites, the data availability, and the computation time for site-pairing.

#### 5.3.1 Distance Distribution of Sites

We discuss the respective distance distributions of the earthquake and tsunami primary-backup site pairs to confirm how close are the Discreet method and the BB method distributions.

In terms of the site pairs for earthquakes, Fig. 7 presents a histogram of the distances between the primary site and the backup site. Figure 8 shows a statistical comparison of the site distances for the earthquake. Site-pairs are commonly used for near field, far field, and coastal earthquakes in the disaster injector of the RDR simulation. There are three attributes to observe in these figures: the average distance, maximum distance, and minimum distance. A large average distance is expected to protect data regardless of the disaster type. A large maximum distance is expected to avoid wiping out all data when almost all sites are damaged. A large minimum distance is expected to delay the

---

**Table 1** Disaster properties.

| #  | Type                  | Hypocenter \([x, y, z]\) | Magnitude |
|----|-----------------------|---------------------------|-----------|
| 1  | Earthquake N (near field) | (15, 10, -40)            | 7.4       |
| 2  | Earthquake C (under coast line) | (30, 10, -40)            | 7.8       |
| 3  | Earthquake F (far field)   | (40, 10, -40)            | 8.1       |
| 4  | Tsunami by earthquake     | (40, 10, -40)            | 8.2       |

---

Intel and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.
onset of data loss with growing disaster degree.

The BB method and the Discreet method have similar narrowly distributed histograms and similar statistics. The standard deviations of both methods are small. The minimum site distances among all pairs are large. The Discreet method has a smaller standard deviation than the BB method, although the average is smaller. In contrast to these two methods, the Greedy method and the Random method have wide distributions. The difference between the Greedy method and the Random method is the average of site distances. Because the Greedy method selects pairs with the largest site distance from the remaining sites, the average is larger than that of the Random method. However, the average of the Greedy method is less than the BB method.

In terms of the tsunami site pairs, Fig. 9 shows a scatter diagram of site distances from the coast. The horizontal axis shows the distance between the primary site and the coast. The vertical axis shows the distance between the corresponding backup site and the coast. In the graph, the top right corner shows the safest pair and the bottom left corner shows the least safe pair.

The graphs show that the Random method has no correlation. It seems clear that the Greedy method has a strong positive correlation except for two irregular points. These two irregular points are generated by the adjustment of the last remaining site because the number of sites is an odd number. This positive correlation signifies that a part of the pairs has extremely strong disaster-resilience but another part of the pairs has extremely week disaster-resilience.

The BB method and the Discreet method have a strong negative correlation. The BB method and the Discreet method have the same shape because the primary-backup site-pairs generated by the BB method and the Discreet method became the same except for the direction of backups among some site triplets, e.g. A→B→C→A vs. A→C→B→A. This negative correlation means that all sites have fairly reasonable disaster-resilience. These results indicate that the BB method and the Discreet method will surely improve data availability.

5.3.2 Data Availability

Next we discuss the data availability, which is the amount of
data that survives the respective disasters.

Figure 10 presents the available data ratios, the percentages of data that survive each disaster when the total number of sites is 135. Each value is an average of 500 earthquake simulations. The error bar represents the standard deviation of the available data ratios.

The available data ratio in the Random method is around 75%, which is the expected value when half of the sites are damaged. This is the baseline value.

The results show that the BB method and the Discreet method are effective for the four types of disaster. In the tsunami case, both methods give the biggest improvement compared with the Random method, with each method raising the data availability by 27.3 points. The reason for the equal degree of improvement is that both methods generate the same primary-backup site-pairs in the tsunami case, as described in the previous subsection.

The Greedy method is effective for coastal and far field earthquakes, but is ineffective for near field earthquake and tsunami disasters. The Greedy method sometimes yields worse results than the Random method. The Greedy method searches for the lowest-risk pairs of sites, then searches for the second lowest-risk pairs of sites, then for the third lowest-risk pairs, and so on. The method finally selects remaining pairs of close sites in the center of the field in the earthquake risk simulation. For this reason, the Greedy method is ineffective for near field earthquakes. Similarly, the Greedy method finally selects remaining pairs of sites near the coast in the tsunami risk simulation. For this reason, the Greedy method is ineffective for tsunami disasters.

Although the BB method gives the optimal solution for the objective function, it does not always achieve the best available data ratio. This is because we use a common risk indicator not focused on a specific disaster type in this paper. As described in Sect. 4.1.1, the reason for this is that nobody can accurately predict the hypocenter or magnitude of a future earthquake.

5.3.3 Computation Time for Site Pairing

We discuss the computation time for site pairing. As the site pairs are decided by the risk indicator, we discuss two cases: the earthquake case and the tsunami case. We basically use the same field conditions as in Fig. 6, except for the number of sites.

Figure 11 shows the computation time as a function of the number of sites when the earthquake risk indicator is used. Both axes are log scale. The computation time of the BB method increases rapidly with increasing number of sites. When the number of sites is 1000, the computation time of the BB method is about 1 hr. However, the computation time for the Greedy and Discreet methods grows at a slower rate. When the number of sites is 1000, the computation times of both methods are a few seconds.

Figure 12 presents the computation time as a function of the number of sites when the tsunami risk indicator is used. The computation time of the BB method is worse than the earthquake case. This is because the computation time of the BB method sometimes depends on the distribution of the objective function coefficients. On the other hand, for the Greedy and Discreet methods there is little change in the computation time from the earthquake case. This is because the computation time for these methods does not depend on the coefficient distribution.

Under more complex conditions such as each of the 1000 sites does not have equal but varying amounts of used
and unused capacity, it is difficult to complete calculations within a couple of weeks using the BB method. However, the computation time of the Discreet method does not change even under such complex conditions.

The results in this and previous subsection show that the Discreet method can select an appropriate backup site to mitigate disaster risk within an extremely short computation time.

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 [13] distributes data over a set of nodes (i.e., storage hosts) based on a consistent hashing technique [14]. 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 [15] 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 [16], Ceph [17], and XtreemFS [18] have georeplication 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 works have produced a disaster-resilient information system [19], [20]. 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 works of fast algorithms to solve IPP are described. Two general-purpose algorithms for IPP are presented because no heuristic algorithm has been proposed for the IPP form in this paper, although the form is similar to the existing Multiple Knapsack Problem [21].

The Branch-and-Bound method (BB method) [7] is a well-known general purpose algorithm. It 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 Dynamic Programming method [22] is also a well-known general purpose algorithm. It 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 [23], which solves the shortest path problem, another form of IPP.

7. Conclusions

As described in this paper, we propose a Discreet method to match safe site-pairs within a short computation time for risk-aware data replication. The Discreet method avoids pairs of primary-backup with either too much or too little safety. We evaluated its data availability for the cases of near-field earthquakes, earthquakes under the coast line, far-field earthquakes, and tsunami disasters. The results show that the Discreet method achieves equivalent data availability to that of the existing Branch-and-Bound method. Indeed, the Discreet method computation time is less than 1/1000 that of the Branch-and-Bound method. We conclude that the Discreet method is useful and that it makes risk-aware data replication more practical for massively multiple sites.

Acknowledgments

The authors thank Masatoshi Shimbori and Hiroshi Ichinomiya for their research computing support, Simon Greaves for his help with English, 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] R.H. Patterson, S. Manley, M. Federwisch, D. Hitz, S. Kleiman, and S. Owara, “SnapMirror: File-system-based asynchronous mirroring for disaster recovery,” Proc. the 1st USENIX Conference on File and Storage Technologies (FAST), Monterey, California, USA, pp.117–129, Jan. 2002.

[2] EMC Education Services, “Remote Replication,” in Information Storage and Management, 2nd ed., John Wiley & Sons, pp.289–310,
[3] Hitachi Data Systems, “Disaster Recovery Issues and Solutions (a white paper),” http://www.hds.com/assets/pdf/wp_117_02_disaster_recovery.pdf, accessed Dec. 19, 2014.

[4] M. Kobayashi, “Experience of Infrastructure Damage Caused by the Great East Japan Earthquake and Countermeasures against Future Disasters,” IEEE Commun. Mag., vol.52, Issue 3, pp.23–29, March 2014.

[5] S. Matsumoto, T. Nakamura, and H. Muraoka, “Risk-aware Data Replication to Massively Multi-sites against Widespread Disasters,” Proc. the Second Asian Conference on Information Systems (ACIS), Phuket, Thailand, pp.34–34, Oct. 2013.

[6] S. Matsumoto, T. Nakamura, and H. Muraoka, “Risk-aware Data Replication to Massively Multi-sites against Widespread Disasters,” Rangsit Journal of Information Technology, vol.1, no.2, pp.22–28, Dec. 2013.

[7] E.L. Lawler and D.E. Wood, “Branch-and-Bound Methods: A survey,” Operations Research July/August, vol.14, no.4, pp.699–719, July-Aug. 1966.

[8] J.B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” Proc. the American Mathematical Society, vol.7, no.1, pp.48–50, Feb. 1956.

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

[10] S. Koshimura, T. Oie, H. Yanagisawa, and F. Imamura, “Developing Fragility Functions for Tsunami Damage Estimation Using Numerical Model and Post-Tsunami Data from Banda Aceh, Indonesia,” Coastal Engineering Journal, vol.51, no.3, pp.243–273, Sept. 2009.

[11] D. Tatsumi and T. Tomita, “Real-time Tsunami Inundation Prediction Using Offshore Tsunami Observation,” Proc. the 32nd International Conference on Coastal Engineering, Shanghai, China, vol.1, no.32, June 2010.

[12] K. Abe, “Estimate of tsunami run-up heights from earthquake magnitudes,” in Tsunami: Progress in Prediction, Disaster Prevention and Warning, Springer Netherlands, vol.44, no.1, pp.21–35, 1995.

[13] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazon’s highly available key-value Store,” Proc. the 21st ACM SIGOPS Symposium on Operating Systems Principles (SOSP), Skamania Lodge, Washington, USA, pp.205–220, Oct. 2007.

[14] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, “Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web,” Proc. the 29th Annual ACM Symposium on Theory of Computing (STOC), pp.654–663, May 1997.

[15] A. Lakshman and M. Malik, “Cassandra: A de-centralized structured storage system,” ACM SIGOPS Operating Systems Review, vol.44, no.2, pp.35–40, April 2010.

[16] D. Muntimadugu, Gluster File System 3.3.0 Administration Guide Using Gluster File System, http://www.gluster.org/wp-content/uploads/2012/05/Gluster_File_System-3.3.0-Administration_Guide-en-US.pdf, accessed Dec. 19, 2014.

[17] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, and C. Malizahn, “Ceph: a scalable, high-performance distributed file system,” Proc. the 7th symposium on Operating systems design and implementation (OSDI), Seattle, Washington, USA, pp.307–320, Nov. 2006.

[18] The XtreemFS Installation and User Guide Version 1.5.x, http://www.xtreemfs.org/xtfs-guide-1.5.pdf, accessed Dec. 19, 2014.

[19] S. Ferdousi, F. Dikbiyik, M.F. Habib, and B. Mukherjee, “Disaster-Aware Data-Center and Content Placement in Cloud Networks,” Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Kattankulathur, India, pp.1–3, Dec. 2013.

[20] S.S. Savas, F. Dikbiyik, M.F. Habib, and B. Mukherjee, “Disaster-Aware Service Provisioning by Exploiting Multipath Routing with Manycasting in Telecom Networks,” Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Kattankulathur, India, pp.1–3, Dec. 2013.

[21] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, John Wiley & Sons, 1990.

[22] R. Bellman, “The Theory of Dynamic Programming,” Bulletin of the American Mathematical Society, vol.60, pp.503–515, 1954.

[23] R. Bellman, “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 are storage system, file system, and operating system.

Shinya Matsumoto received B.E. (2004) and M.E. (2006) degrees from Nagoya University, Japan. He currently works in the Data Storage Systems Department of Yokohama Research Laboratory, Hitachi, Ltd., Japan. His current research interests include operating systems, file systems, and data management in storage systems.

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.