



## Balancing Reliability and Cost in Cloud-RAID Systems with Fault-Level Coverage

**Lavanya Mandava**

Department of Electrical and Computer Engineering  
University of Massachusetts, Dartmouth, MA, USA  
E-mail: lmandava@umassd.edu

**Liudong Xing**

Department of Electrical and Computer Engineering  
University of Massachusetts, Dartmouth, MA, USA  
*Corresponding author:* lxing@umassd.edu

(Received May 13, 2019; Accepted June 1, 2019)

### Abstract

Based on redundancy techniques, cloud-RAIDs (Redundant Array of Independent Disks) offer an effective storage solution to achieve high data reliability. Their performance however can be greatly hindered by the fault-level coverage (FLC) behavior, where an uncovered disk fault may crash the entire system in spite of adequate redundancy remaining. Moreover, different choices of cloud disk providers lead to designs with different overall reliability and cost. Thus, in this paper we formulate and solve optimization problems, which determine the combination of cloud disks (from different providers) maximizing the cloud-RAID system reliability or minimizing the total cost. The cloud-RAID reliability is analyzed using a combinatorial and analytical modeling method while considering effects of the FLC behavior. Multiple case studies are performed to demonstrate the considered optimization problems and proposed solution methodology.

**Keywords-** Cloud storage, Cost, Fault level coverage, Optimization, Reliability.

### 1. Introduction

Recent advances in big data, Internet of Things, cyber-physical systems have led to great demands on data storage (Atat et al., 2018; Wang et al., 2018; Wang and Alexander, 2019). These demands engender high needs to use cloud storage services as the backbone of those technologies. As a continuously growing paradigm, a cloud storage system is a model with data being saved in logical space while the actual disks may span several physical servers managed by different cloud service providers (Deng et al., 2010; Erl et al., 2013). As users often expect to access their data from the cloud storage system anytime and anywhere, any service interruption or failure can pose negative impacts on the reputation and business of the cloud service provider. Thus, it is significant to enhance the overall data reliability through fault-tolerance techniques. Based on redundancy techniques (Jin et al., 2009; Bausch, 2014), cloud-RAIDs (Redundant Array of Independent/Inexpensive Disks) provide one such solution to achieve high data reliability (Fitch and Xu, 2013; Zhang et al., 2013; Liu and Xing, 2015). Their performance however may be greatly affected by the imperfect coverage behavior (Xing, 2005; Myers, 2010; Li and Mao, 2016), where due to an imperfect fault recovery mechanism, a disk fault may not be adequately or timely detected, located and isolated after its occurrence. The consequence from this undetected or uncovered fault is it may propagate to other system components, causing extensive damages and even crashing the entire system. Studies performed in (Amari et al., 1999) showed that an increase in the redundancy level may not necessarily enhance the reliability of systems subject to the imperfect fault coverage monotonically (after certain level, a further increase in redundancy can



actually reduce the system reliability). Hence, it is crucial to address the imperfect fault coverage behavior in the system reliability modeling, analysis and optimization activities.

Some efforts have been expended in optimizing the cloud storage systems. Examples include an algorithmic approach (Goyal and Kant, 2014) suggested to optimize cloud storage for improving data accessing and storage performance. In Yahyaoui and Moalla (2016), the optimization of file storage service was considered through classifying customer files based on their usage rates to save storage space in the cloud. In Liu et al. (2018), a heterogeneous cloud storage model was optimized achieving the tradeoff between the system storage and repair costs. In Li et al. (2016), the structure and performance of an object-based cloud storage system were optimized for processing data files with different sizes. In Fu et al. (2016), a block replica placement method was suggested for optimizing performance of small files accessing in the cloud storage system. In Cha and Kim (2018), a software-defined storage using a combination of on-premises and public cloud storage was introduced to optimize the cloud storage for improving the I/O performance. In Mansouri et al. (2017), an optimal offline algorithm was designed to optimize the price difference between cloud data storage and cloud network services. In Al-Abbasi and Aggarwal (2018), a framework for erasure-coded storage systems was proposed to quantify and optimize the mean latency over the choice of cloud storage servers and auxiliary bound parameters. The above-mentioned methods neither addressed the imperfect fault coverage behavior, nor considered the cloud provider/disk selection problem that determines the optimal combination of cloud disks balancing the cloud-RAID system reliability and cost. The selection problem belongs to the reliability allocation problem, which has been proven to be NP-hard (Chern, 1992; Todinov, 2006).

Different models have been suggested for addressing the imperfect fault coverage in reliability analysis of systems in diverse applications. Typical examples include the element level coverage (ELC) model (also known as single-fault model), the fault-level coverage (FLC) model (also known as multi-fault model), and the performance dependent coverage model (Levitin and Amari, 2008; Mandava et al., 2016; Mandava and Xing, 2017). In recent work (Mandava and Xing, 2019), the cloud disk selection problem was first considered for the cloud-RAID system subject to the ELC, where the performance of the system recovery mechanism depends on the occurrence of each individual disk fault and thus the fault coverage probability of a disk does not rely on statuses of other disks in the system. However, in a load-sharing or work-sharing environment, it is more practical that the fault coverage probability relies on the number of disk faults that have happened within a certain recovery window, which can be modeled by the FLC (Amari et al., 2008). In this paper, we make advancements by formulating and solving the cloud disk selection problem for cloud-RAID storage systems undergoing the FLC. Both unconstrained and constrained optimization problems are considered. Solutions to the constrained problems provide a balance between the overall reliability and cost when configuring a cloud-RAID system.

The rest of the paper is arranged as follows. Section 2 formulates the considered optimization problems. Section 3 describes examples of cloud-RAID storage systems. Section 4 presents the reliability analysis method for the cloud-RAID systems undergoing the FLC. Section 5 demonstrates solutions to the considered optimization problems through multiple case studies. Section 6 concludes the work and gives directions of future research.



## 2. Formulation of Cloud Disk Selection Problems

In (1) an unconstrained optimization problem is formulated with the objective to minimize the overall cloud-RAID system unreliability, denoted as  $UR(t)$  (i.e., maximize the system reliability  $1-UR(t)$ ).

$$\text{minimize } UR(t) \quad (1)$$

In (2) and (3), two different constrained optimization problems are formulated to balance the system reliability and cost. Specifically, the aim of problem (2) is to minimize the cloud-RAID system unreliability,  $UR(t)$  while satisfying constraint on the total system cost denoted by  $C^*$ . The aim of problem (3) is to minimize the total system cost  $C$  while satisfying constraint on the cloud-RAID system unreliability denoted by  $UR^*$ .

$$\text{minimize } UR(t) \quad \text{s.t. } C \leq C^* \quad (2)$$

$$\text{minimize } C \quad \text{s.t. } UR(t) \leq UR^* \quad (3)$$

$UR(t)$  in (1)-(3) is evaluated using a combinatorial and analytical modeling method that is presented in Section 4.  $C$  is evaluated as summation of costs of all the chosen disks to configure the cloud-RAID system.

## 3. Example Cloud-RAID 5 Storage Systems

There are seven levels in the conventional RAID architecture employing different redundancy techniques (except level one without any redundancy) (Jin et al., 2011). The RAID level 5 using the distributed single parity code is selected in this work to illustrate the proposed methodology and optimization. Specifically, data to be saved in RAID 5 are divided into non-overlapped blocks. These blocks are striped across different disks that form the array (Patterson et al., 1989). The parity stripes are also distributed across those disks in the array. The RAID 5 is able to tolerate any single disk failure that can be detected and located successfully. Specifically, if any disk becomes malfunctioning or unavailable, stripes on this disk can be restored using the parity stripe and data stripes from the remaining disks (in particular through performing the exclusive OR operation if an even parity code is utilized). In the case of disk drives within the array having different capacities, the usable capacity by the whole storage array is decided by the disk with the smallest storage capacity.

Figure 1 shows the architecture of an example cloud-RAID 5 system with three disk drives coming from different cloud service providers. Users' data ( $A, B, C$ ) are divided into stripes (for example,  $A_1$  and  $A_2$  for  $A$ ). Both data and parity stripes are distributed among the three disk drives. The system is essentially a 2-out-of-3: G model, meaning that the system is good or reliable if at least two out of the three disks are functioning correctly.

Figure 2 shows architecture of a larger cloud-RAID 5 with five different disks. It is a 4-out-of-5: G model, meaning that the system is good if at least four out of the five disks are functioning correctly.

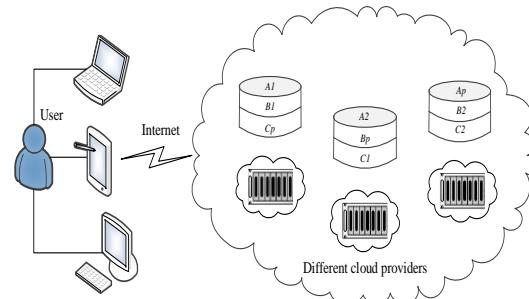


Figure 1. 2-out-of-3 cloud-RAID 5

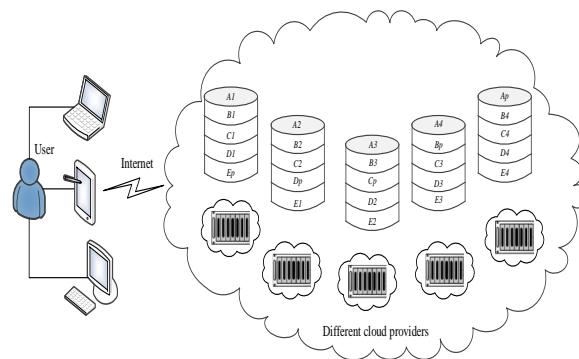


Figure 2. 4-out-of-5 cloud-RAID 5

#### 4. Combinatorial Method for Analyzing Cloud-RAID 5 with FLC

Under the FLC, the fault coverage probability relies on the number of disk faults happening to a particular group (disk array) within a certain recovery window  $\tau$  (Levitin and Amari, 2007; Myers and Rauzy, 2008). Specifically, a set of fault coverage factors  $c_i$  is evaluated for a specific disk group. For example, the first disk fault is covered with probability  $c_1$ , the second disk fault is covered with probability  $c_2$ , and so on. By definition, the coverage factor involving no disk faults  $c_0$  is always 1. Formula (4) presents an example method of evaluating  $c_i$  for a system with  $n$  identical disk drives, where  $i$  denotes the fault number and  $\lambda$  denotes the constant failure rate of each disk.

$$c_i = e^{-(n-i)\lambda\tau} \quad (4)$$

In the case of non-identical disks with different failure rates (in general, failure time distributions), formula (4) needs to be modified to consider a different reliability evaluation for each disk based on its time-to-failure distribution function. Specifically, let  $c_{id}$  denote the coverage probability associated with the  $i$ -th fault happening to disk  $d$  in the cloud-RAID system. As mentioned in Section 3, the cloud-RAID 5 system can tolerate any single disk failure if the fault is detected and located successfully. This first disk failure ( $i=1$ ) is covered with the coverage probability,  $c_{1d}$ . For example, for the example 2-out-of-3 cloud-RAID system, if disk 1 fails first, the coverage probability  $c_{11}$  can be evaluated as (5); if disk 2 fails first,  $c_{12}$  in (5) is used; if disk 3 fails first,  $c_{13}$  in (5) is used.

$$c_{11} = e^{-(\lambda_2 + \lambda_3)\tau}, \quad c_{12} = e^{-(\lambda_1 + \lambda_3)\tau}, \quad c_{13} = e^{-(\lambda_1 + \lambda_2)\tau} \quad (5)$$

To consider effects of FLC, the binary decision diagram (BDD)-based method (Xing and Amari, 2015; Xing et al., 2019) can be applied with an extension of inserting corresponding coverage factor  $c_{id}$  onto the relevant paths in the system BDD model. The following major steps are involved in the extended BDD-based reliability analysis:

- 1) **Variable Ordering:** assign indexes to input disk variables using the numerical order.
- 2) **BDD Generation:** generate the BDD model without considering FLC by recursively applying manipulation rules in (6), which are used to combine two sub-BDDs into one BDD.

$$\begin{aligned} g \diamond h &= ite(x, G_1, G_0) \diamond ite(y, H_1, H_0) \\ &= \begin{cases} ite(x, G_1 \diamond H_1, G_0 \diamond H_0) & index(x) = index(y) \\ ite(x, G_1 \diamond h, G_0 \diamond h) & index(x) < index(y) \\ ite(y, g \diamond H_1, g \diamond H_0) & index(x) > index(y) \end{cases} \end{aligned} \quad (6)$$

$g$  and  $h$  in (6) are Boolean functions in the if-then-else ( $ite$ ) format representing the two sub-BDDs to be combined ( $x$  and  $y$  are their root nodes).  $\diamond$  denotes a logical OR/AND operation. For applying the rules, indexes of  $x$  and  $y$  are compared. In the first case, their indexes are identical meaning they are the same variable. In this case, the operation is performed between their child nodes. In the second and third cases,  $x$  and  $y$  have different indexes meaning they are variables of different disks. In this case, the variable with a smaller index becomes the root node of the combined BDD, and the logic operation is performed between each child node of the smaller index node and the entire sub-BDD rooted at the larger index node.

- 3) **Coverage Factor Nodes Insertion:** node associated with each relevant coverage factor  $c_{id}$  is inserted onto the relevant operational path (i.e., path from root node to sink node ‘0’) in the BDD model generated in step 2 for the  $i$ -th failure happening to disk  $d$ .

For the example cloud-RAID 5 model considered in this paper, all operational paths involve either no failures or 1 disk failure. Thus, for paths involving a single disk failure happening to disk  $d$ , node  $c_{1d}$  is inserted; for paths involving no disk failures, node  $c_0$  is simply inserted. For systems that can tolerate 2 disk failures (e.g., the cloud-RAID 6 model in Mandava and Xing, 2017), there exist operational paths involving no failures, 1 disk failure, and 2 disk failures. For the path involving 2 disk failures, e.g., happening to disk  $d1$  first and then disk  $d2$ , nodes  $c_{1d1}$  and  $c_{2d2}$  are inserted onto the path.

- 4) **BDD Evaluation:** the cloud-RAID system unreliability  $UR(t)$  is evaluated as summation of probabilities of all disjoint paths from the root node of the BDD constructed at step 3 to sink node ‘1’. Similarly, the reliability of the cloud-RAID system is the sum of probabilities of all disjoint paths from the root node to sink node ‘0’.

Each single path probability is evaluated as the product of disk reliability  $p_d$  (if 0-edge appearing on the path) or unreliability  $1 - p_d$  (if 1-edge appearing on the path) for all disks  $d$  involved on the path.

Reliability  $p_d$  of each single disk  $d$  is either given as an input parameter or can be derived from the time-to-failure distribution parameter(s). For example, if disk  $d$  has the exponential time-to-failure distribution with parameter  $\lambda_d$  then  $p_d = e^{(-\lambda_d * t)}$ ; if disk  $d$  has the Weibull time-to-failure distribution with scale parameter  $\lambda_d$  and shape parameter  $\beta_d$ , then  $p_d = e^{-(t * \lambda_d)^\beta_d}$ .

#### 4.1 Analysis of the 2-out-of-3 Cloud-RAID 5

Figure 3 illustrates the BDD model constructed for the 2-out-of-3 cloud-RAID 5 with FLC. Each non-sink node has two outgoing edges: the solid edge (or 0-edge) represents that the disk represented by the node is operating and the dashed edge (or 1-edge) represents the disk is failed. To save space, only paths to sink node ‘0’ representing the entire system is working are shown. Coverage factors are inserted into the corresponding paths as described in step 3 of the algorithm. For example, in the second left-most path where only disk 3 fails,  $c_{13}$  is inserted onto that path.

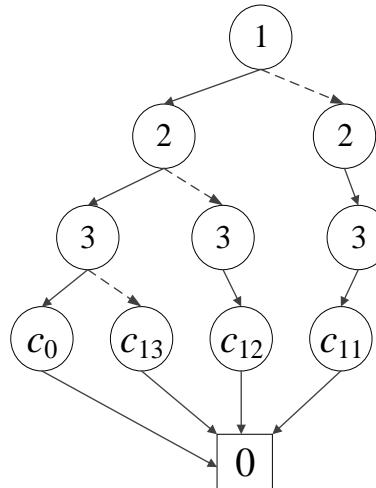


Figure 3. BDD of the example 2-out-of-3 cloud-RAID 5 with FLC

Based on the BDD model in Figure 3, the unreliability of the example 2-out-of-3 cloud-RAID 5 is evaluated as

$$UR(t) = 1 - [(1 - p_1)p_2p_3c_{11} + p_1(1 - p_2)p_3c_{12} + p_1p_2(1 - p_3)c_{13} + p_1p_2p_3c_0] \quad (7)$$

#### 4.2 Analysis of 4-out-of-5 Cloud-RAID 5

Figure 4 shows the BDD model generated for the example 4-out-of-5 cloud-RAID 5 with FLC. Similar to Figure 3, only paths to sink node ‘0’ representing the system is working are shown. Based on the generated BDD model, the system unreliability can be evaluated using (8)

$$UR(t) = 1 - [(1 - p_1)p_2p_3p_4p_5c_{11} + p_1(1 - p_2)p_3p_4p_5c_{12} + p_1p_2(1 - p_3)p_4p_5c_{13} + p_1p_2p_3(1 - p_4)p_5c_{14} + p_1p_2p_3p_4(1 - p_5)c_{15} + p_1p_2p_3p_4p_5c_0] \quad (8)$$

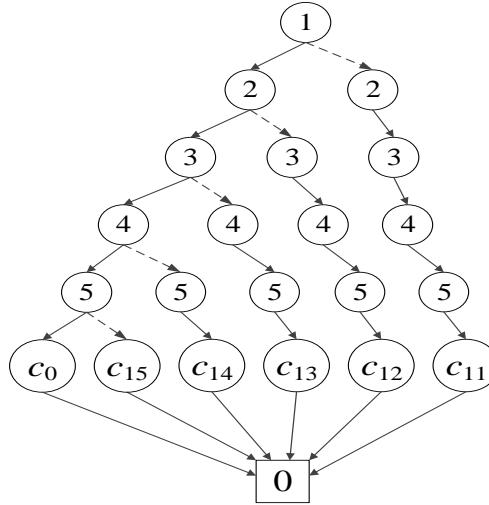


Figure 4. BDD of the example 4-out-of-5 cloud-RAID 5 with FLC

## 5. Optimization

Consider a cloud-RAID system using  $n$  disk drives selected from  $w$  available choices (cloud providers). The number of possible combinations of those disk drives is  $\frac{(n+w-1)!}{(w-1)!n!}$ . For example, when  $n=3$  and  $w=2$  (i.e., a 2-out-3 cloud-RAID is configured with disks provided by two cloud providers ( $v_1, v_2$ )), there are 4 distinct combinations:  $(v_1, v_1, v_1)$ ,  $(v_1, v_1, v_2)$ ,  $(v_1, v_2, v_2)$ ,  $(v_2, v_2, v_2)$ . The ordering does not matter, e.g.,  $(v_2, v_1, v_1)$  and  $(v_1, v_2, v_1)$  are considered the same combination.

For values of  $n$  and  $w$  used in the practical implementation of the cloud-RAID system, the brute force approach can be sufficiently applied, which searches all possible combinations to find the optimal solution for problems formulated in (1)-(3). Theoretically, for large values of  $n$  and  $w$ , heuristic approaches e.g., the genetic algorithm (Tannous et al., 2011; Boddu and Xing, 2013; Bhunia et al., 2017) and meta-heuristic approaches (Dahiya et al., 2019) may be applied to solve those problems.

Table 1. Input parameters of cloud disk drives (providers)

Disk providers	$\lambda_d$ (per hour)	$C_d$ (per hour)
$v_1$	0.0003	\$0.0014
$v_2$	0.00005	\$0.0042
$v_3$	0.00003	\$0.0056
$v_4$	0.0001	\$0.0028
$v_5$	0.00001	\$0.007

Table 1 presents parameters of cloud disk drives used in four case studies performed in the following subsections, including constant failure rate  $\lambda_d$  (the exponential distribution is assumed)

and cost  $C_d$  for disks from five providers ( $v_1, v_2, v_3, v_4, v_5$ ). The cost  $C_d$  is assigned based on recent prices of several top cloud service providers in market, like AWS, Dropbox, Google, iCloud (Amazon, 2019; Dropbox, 2019; Google, 2019; iCloud, 2019).

### 5.1 Case Study 1: 2-out-of-3 Cloud-RAID 5 with 3 Disk Providers

Providers  $v_1, v_2, v_3$  are used in this case study. Table 2 lists all the ten possible combinations and corresponding system unreliability and cost for three different sets of  $t$  and  $\tau$  values. As the mission time proceeds (comparing columns 3 and 4) or the recovery window time increases (comparing columns 4 and 5), the system unreliability increases. For problem (1), the optimal combination is  $(v_3, v_3, v_3)$  for all the three sets of  $t$  and  $\tau$  values listed, respectively with the minimum system unreliability of 0.002584, 0.117226, and 0.117724. This is intuitive since among the three available providers,  $v_3$  has the lowest failure rate (the most reliable disks). For problem (2) with  $C^* = \$10$ , the optimal solution is  $(v_1, v_2, v_2)$  for all the three sets of  $t$  and  $\tau$  with cost of \\$9.8. For problem (3) under  $\tau = 3\text{hrs}$ ,  $t=1000\text{hrs}$ , and  $UR^* = 0.01$ , the optimal solution is  $(v_2, v_2, v_2)$  with the system cost of \\$12.6 and unreliability of 0.006943.

Table 2. Results of 2-out-of-3 with 3 cloud providers

Combinations	Cost	$UR(t)$ ( $\tau = 3\text{hrs}, t=1000\text{hrs}$ )	$UR(t)$ ( $\tau = 3\text{hrs}, t=8000\text{hrs}$ )	$UR(t)$ ( $\tau = 24\text{hrs}, t=8000\text{hrs}$ )
$v_1, v_1, v_1$	\$4.2	0.167472	0.976844	0.977125
$v_1, v_1, v_2$	\$7	0.086335	0.881304	0.882147
$v_1, v_1, v_3$	\$8.4	0.078922	0.862127	0.863044
$v_1, v_2, v_2$	\$9.8	0.026569	0.510740	0.511890
$v_1, v_2, v_3$	\$11.2	0.021109	0.436358	0.437420
$v_1, v_3, v_3$	\$12.6	0.015827	0.350895	0.351813
$v_2, v_2, v_2$	\$12.6	0.006943	0.254535	0.255467
$v_2, v_2, v_3$	\$14	0.005150	0.203108	0.203892
$v_2, v_3, v_3$	\$15.4	0.003692	0.156288	0.156923
$v_3, v_3, v_3$	\$16.8	0.002584	0.117226	0.117724

### 5.2 Case Study 2: 2-out-of-3 Cloud-RAID 5 with 5 Disk Providers

Table 3 presents the optimization results for the 2-out-of-3 cloud-RAID 5 model considering five disk providers  $v_1, v_2, v_3, v_4, v_5$  for problems (1)-(3) assuming  $t=1000\text{hrs}$ ,  $\tau=3\text{hrs}$ .

Table 3. Optimal results of 2-out-of-3 with 5 disk providers

Optimization Problem	Optimal Solution	Cost $C$	Unreliability $UR(t)$
$\min UR(t)$	$v_5, v_5, v_5$	\$21	0.000297
$\min UR(t) \text{ s.t. } C \leq C_1^* = \$8$	$v_1, v_4, v_4$	\$7	0.053971
$\min UR(t) \text{ s.t. } C \leq C_2^* = \$12$	$v_2, v_2, v_4$	\$11.2	0.011272
$\min UR(t) \text{ s.t. } C \leq C_3^* = \$17$	$v_4, v_5, v_5$	\$16.8	0.001985
$\min C \text{ s.t. } UR(t) \leq UR_1^* = 0.02$	$v_2, v_4, v_4$	\$9.8	0.017553
$\min C \text{ s.t. } UR(t) \leq UR_2^* = 0.005$	$v_3, v_4, v_5$	\$15.4	0.004021
$\min C \text{ s.t. } UR(t) \leq UR_3^* = 0.001$	$v_3, v_5, v_5$	\$19.6	0.000685

### 5.3 Case Study 3: 4-out-of-5 Cloud-RAID 5 with 3 Disk Providers

Assume disk providers  $v_1, v_2, v_3$  are used in this case study. Table 4 lists all the 21 possible combinations and corresponding system unreliability and cost for three different sets of  $t$  and  $\tau$  values.

Table 4. Results of 4-out-of-5 with 3 cloud providers

Combinations	Cost	$UR(t)$ ( $\tau = 3\text{hrs}$ , $t=1000\text{hrs}$ )	$UR(t)$ ( $\tau = 3\text{hrs}$ , $t=8000\text{hrs}$ )	$UR(t)$ ( $\tau = 24\text{hrs}$ , $t=8000\text{hrs}$ )
$v_1, v_1, v_1, v_1, v_1$	\$7	0.387952	0.999687	0.999695
$v_1, v_1, v_1, v_1, v_2$	\$9.8	0.299055	0.998117	0.998154
$v_1, v_1, v_1, v_1, v_3$	\$11.2	0.290933	0.997802	0.997844
$v_1, v_1, v_1, v_2, v_2$	\$12.6	0.209197	0.989269	0.989423
$v_1, v_1, v_1, v_2, v_3$	\$14	0.200988	0.987493	0.987667
$v_1, v_1, v_1, v_3, v_3$	\$15.4	0.192770	0.985424	0.985620
$v_1, v_1, v_2, v_2, v_2$	\$15.4	0.125052	0.944249	0.944769
$v_1, v_1, v_2, v_2, v_3$	\$16.8	0.117365	0.935212	0.935787
$v_1, v_1, v_2, v_3, v_3$	\$18.2	0.109725	0.924718	0.925353
$v_1, v_1, v_3, v_3, v_3$	\$19.6	0.102138	0.912532	0.913231
$v_1, v_2, v_2, v_2, v_2$	\$18.2	0.057174	0.762230	0.763337
$v_1, v_2, v_2, v_2, v_3$	\$19.6	0.050973	0.725694	0.726846
$v_1, v_2, v_2, v_3, v_3$	\$21	0.044907	0.683638	0.684821
$v_1, v_2, v_3, v_3, v_3$	\$22.4	0.038985	0.635247	0.636443
$v_1, v_3, v_3, v_3, v_3$	\$23.8	0.033216	0.579589	0.580772
$v_2, v_2, v_2, v_2, v_2$	\$21	0.021669	0.532058	0.533452
$v_2, v_2, v_2, v_2, v_3$	\$22.4	0.018425	0.485857	0.487215
$v_2, v_2, v_2, v_3, v_3$	\$23.8	0.015452	0.437715	0.439018
$v_2, v_2, v_3, v_3, v_3$	\$25.2	0.012761	0.388352	0.389580
$v_2, v_3, v_3, v_3, v_3$	\$26.6	0.010365	0.338793	0.339928
$v_3, v_3, v_3, v_3, v_3$	\$28	0.008277	0.290459	0.291487

For problem (1), the optimal combination is  $(v_3, v_3, v_3, v_3, v_3)$  for all the three sets of  $t$  and  $\tau$  values, respectively with the minimum system unreliability of 0.008277, 0.290459, and 0.291487. This is intuitive since among the three available providers,  $v_3$  has the most reliable disks; without any cost constraints,  $v_3$  should be selected for all the five disks. For problem (2) with  $C^* = \$20$ , the optimal solution is  $(v_1, v_2, v_2, v_2, v_3)$  for all the three sets of  $t$  and  $\tau$  with the system cost of \$19.6. For problem (3) under  $\tau = 3\text{hrs}$ ,  $t=1000\text{hrs}$ , and  $UR^* = 0.07$ , the optimal solution is  $(v_1, v_2, v_2, v_2, v_2)$  with the system cost of \$18.2 and unreliability of 0.057174.

### 5.4 Case Study 4: 4-out-of-5 Cloud-RAID 5 with 5 Disk Providers

Table 5 presents the optimization results for the 4-out-of-5 cloud-RAID 5 model using five disk providers  $v_1, v_2, v_3, v_4, v_5$  for problems (1)-(3) assuming  $t=1000\text{hrs}$ ,  $\tau=3\text{hrs}$ .

Table 5. Optimal results of 4-out-of-5 with 5 disk providers

Optimization Problem	Optimal Solution	Cost $C$	Unreliability $UR(t)$
$\min UR(t)$	$v_5, v_5, v_5, v_5, v_5$	\$35	0.000976
$\min UR(t)$ s.t. $C \leq C_1^* = \$12$	$v_1, v_1, v_4, v_4, v_4$	\$11.2	0.181488
$\min UR(t)$ s.t. $C \leq C_2^* = \$20$	$v_2, v_2, v_2, v_2, v_4$	\$19.6	0.0295
$\min UR(t)$ s.t. $C \leq C_3^* = \$28$	$v_2, v_2, v_3, v_5, v_5$	\$28	0.007563
$\min C$ s.t. $UR(t) \leq UR_1^* = 0.05$	$v_2, v_2, v_4, v_4, v_4$	\$16.8	0.049712
$\min C$ s.t. $UR(t) \leq UR_2^* = 0.009$	$v_2, v_4, v_5, v_5, v_5$	\$28	0.008925
$\min C$ s.t. $UR(t) \leq UR_3^* = 0.004$	$v_2, v_3, v_5, v_5, v_5$	\$30.8	0.003966



## 6. Conclusion and Future Work

Disk drives from different cloud service providers are often characterized with different reliability performance and cost. Different combinations of available disk choices thus lead to different overall cloud storage system reliability and cost. This paper models and solves three types of cloud disk provider selection problems for cloud-RAID storage systems subject to the FLC, including an unconstrained problem minimizing the overall system unreliability, and two constrained problems balancing the overall system reliability and cost. The reliability of the considered cloud-RAID system is analyzed using an extended BDD-based method that has no restriction on time-to-failure distribution types for disks. The solution methodology and proposed optimizations are demonstrated through examples and four case studies.

One direction of our future work is to extend the proposed methodology to optimize multi-state cloud-RAID storage systems subject to performance degradation and the FLC (Mandava et al., 2019). We are also interested in studying the performance dependent coverage model for reliability analysis and optimization of cloud-RAID systems.

### Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

### Acknowledgements

The authors would like to express their sincere thanks to the editor and anonymous reviews for their time and valuable suggestions.

## References

- Al-Abbasi, A.O., & Aggarwal, V. (2018, April). Mean latency optimization in erasure-coded distributed storage systems. *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, (pp. 432-437). IEEE.
- Amari, S.V., Dugan, J.B., & Misra, R.B. (1999). Optimal reliability of systems subject to imperfect fault-coverage. *IEEE Transactions on Reliability*, 48(3), 275-284.
- Amari, S.V., Myers, A.F., Rauzy, A., & Trivedi, K.S. (2008). Imperfect coverage models: status and trends. In *Handbook of Performability Engineering*, (pp. 321-348). Springer, London.
- Atat, R., Liu, L., Wu, J., Li, G., Ye, C., & Yang, Y. (2018). Big data meet cyber-physical systems: a panoramic survey. *IEEE Access*, 6, 73603-73636.
- Bhunia, A.K., Duary, A., & Sahoo, L. (2017). A genetic algorithm based hybrid approach for reliability redundancy optimization problem of a series system with multiple choice. *International Journal of Mathematical, Engineering and Management Sciences*, 2(3), 185-212.
- Boddu, P., & Xing, L. (2013). Reliability evaluation and optimization of series-parallel systems with k-out-of-n: G subsystems and mixed redundancy types. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 227(2), 187-198.
- Cha, J., & Kim, S. (2018, December). Analysis of i/o performance for optimizing software defined storage in cloud integration. In *IEEE 3rd International Conference on Communication and Information Systems*, (pp. 222-226).

- Chern, M.S. (1992). On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, 11(5), 309-315.
- Dahiya, B.P., Rani, S., & Singh, P. (2019). A hybrid artificial grasshopper optimization (hagoa) meta heuristic approach: a hybrid optimizer for discover the global optimum in given search space. *International Journal of Mathematical, Engineering and Management Sciences*, 4(2), 471-488.
- Deng, J., Huang, S.C.H., Han, Y.S., & Deng, J.H. (2010, December). Fault-tolerant and reliable computation in cloud computing. In *IEEE Globecom Workshops*, (pp. 1601-1605). IEEE.
- Erl, T., Puttini, R., & Mahmood, Z. (2013). *Cloud computing concepts, technology & architecture*. The Prentice Hall Service Technology Series, Prentice Hall.
- Fitch, D., & Xu, H. (2013). A RAID-based secure and fault-tolerant model for cloud information storage. *International Journal of Software Engineering and Knowledge Engineering*, 23(05), 627-654.
- Fu, X., Liu, W., Cang, Y., Gong, X., & Deng, S. (2016). Optimized data replication for small files in cloud storage systems. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation.
- Goyal, V., & Kant, C. (2014, February). An effective algorithmic approach for cost optimization in cloud based data center. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, (pp. 630-637). IEEE.
- Jin, T., Xing, L., & Yu, Y. (2011). A hierarchical Markov reliability model for data storage systems with media self-recovery. *International Journal of Reliability, Quality and Safety Engineering*, 18(01), 25-41.
- Jin, T., Yu, Y., & Xing, L. (2009, July). Reliability analysis of RAID systems using repairable k-out-of-n modeling techniques. In *The International Conference on the Interface between Statistics and Engineering, Beijing, China*.
- Levitin, G., & Amari, S.V. (2007). Reliability Analysis of fault tolerant systems with multi-fault coverage. *International Journal of Performability Engineering*, 3(4), 441-451.
- Levitin, G., & Amari, S.V. (2008). Multi-state systems with multi-fault coverage. *Reliability Engineering & System Safety*, 93(11), 1730-1739.
- Li, L., Li, D., Su, Z., Jin, L., & Huang, G. (2016). Performance analysis and framework optimization of open source cloud storage system. *China Communications*, 13(6), 110-122.
- Li, Q., & Mao, C. (2016, October). Considering testing-coverage and fault removal efficiency subject to the random field environments with imperfect debugging in software reliability assessment. In *Proceedings of IEEE 27th International Symposium on Software Reliability Engineering Workshops*, (pp. 257-263). Ottawa, ON, Canada.
- Liu, P., Zheng, L., Yu, Q., & Ye, H. (2018, April). Tradeoff between storage cost and repair cost for cloud storage. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, (pp. 169-173). IEEE.
- Liu, Q., & Xing, L. (2015). Reliability modeling of cloud-RAID-6 storage system. *International Journal of Future Computer and Communication*, 4(6), 415-420.
- Mandava, L., & Xing, L. (2019). Optimizing imperfect coverage cloud-RAID systems considering reliability and cost. *International Journal of Reliability, Quality & Safety Engineering*, Special Issue on Reliability Topics from ICMTEA2018 (accepted).
- Mandava, L., & Xing, L. (2017). Reliability analysis of cloud-RAID 6 with imperfect fault coverage. *International Journal of Performability Engineering*, 13(3), 289-297.

- Mandava, L., Xing, L., & Pan, Z. (2016). Imperfect coverage analysis for cloud-RAID 5. In *Engineering Asset Management*, ed. M. Zuo, L. Ma, J. Mathew, and HZ. Huang, Lecture Notes in Mechanical Engineering, (pp. 207-220). Cham, Springer.
- Mandava, L., Xing, L., Vokkarane, V.M., & Tannous, O. (2019). Reliability analysis of multi-state cloud-RAID with imperfect element-level coverage. In *Reliability Engineering: Theory and Applications Chapter 4* (pp. 61-82). Editors: Ilia Vonta and Mangey Ram, CRC Press/Taylor and Francis.
- Mansouri, Y., Toosi, A.N., & Buyya, R. (2017). Cost optimization for dynamic replication and migration of data in cloud data centers. *IEEE Transactions on Cloud Computing*, pp. 1-1. DOI: 10.1109/TCC.2017.2659728.
- Myers, A. (2010). *Complex system reliability*. Springer Series in Reliability Engineering, 2nd edition (pp. 27-37).
- Myers, A.F., & Rauzy, A. (2008). Assessment of redundant systems with imperfect coverage by means of binary decision diagrams. *Reliability Engineering & System Safety*, 93(7), 1025-1035.
- Patterson, D.A., Chen, P., Gibson, G., & Katz, R.H. (1989, February). Introduction to redundant arrays of inexpensive disks (RAID). In *Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage*, (pp. 112-117). IEEE.
- Tannous, O., Xing, L., Rui, P., Xie, M., & Ng, S.H. (2011, December). Redundancy allocation for series-parallel warm-standby systems. In *2011 IEEE International Conference on Industrial Engineering and Engineering Management*, (pp. 1261-1265). IEEE.
- Todinov, M.T. (2006). Reliability analysis of complex systems based on the losses from failures. *International Journal of Reliability, Quality and Safety Engineering*, 13(02), 127-148.
- Wang, L., & Alexander, C.A. (2019). Big data analytics in healthcare systems. *International Journal of Mathematical, Engineering and Management Sciences*, 4(1), 17-26.
- Wang, W., Xu, P., & Yang, L.T. (2018). Secure data collection, storage and access in cloud-assisted IoT. *IEEE Cloud Computing*, 5(4), 77-88.
- Xing, L. (2005). Reliability modeling and analysis of complex hierarchical systems. *International Journal of Reliability, Quality and Safety Engineering*, 12(06), 477-492.
- Xing, L., & Amari, S.V. (2015). *Binary decision diagrams and extensions for system reliability analysis*. Wiley-Scrivener, MA, ISBN: 978-1-118-54937-7.
- Xing, L., Levitin, G., & Wang, C. (2019). *Dynamic system reliability: modeling and analysis of dynamic and dependent behaviors*. Wiley, ISBN: 978-1-119-50763-5.
- Yahyaoui, H., & Moalla, S. (2016, December). Cloud FC: files clustering for storage space optimization in Clouds. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, (pp. 193-197). IEEE.
- Zhang, R., Lin, C., Meng, K., & Zhu, L. (2013, November). A modeling reliability analysis technique for cloud storage system. In *Proceedings of the 15th IEEE International Conference on Communication Technology*, (pp. 32-36). IEEE.

## Links and Websites

- Amazon (2019). Simple Storage Service (S3), <https://aws.amazon.com/govcloud-us/pricing/s3/>, Accessed in May 2019.
- Bausch, F. (2014, April). Cloud-RAID Concept. <http://blog.fbausch.de/cloudraid-3-concept/>, Accessed in May 2019.



Dropbox (2019). Business Pricing, <https://www.dropbox.com/business/pricing>, Accessed in May 2019.

Google (2019). Google One, <https://one.google.com/about>, Accessed in May 2019.

iCloud (2019). Storage Plans and Pricing, <https://support.apple.com/en-us/HT201238>, Accessed in May 2019.



Original content of this work is copyright © International Journal of Mathematical, Engineering and Management Sciences. All rights reserved.  
Except of uses under a Creative Commons Attribution 4.0 International (CC BY 4.0) license at <https://creativecommons.org/licenses/by/4.0/>