RZ 3827 (# ZUR1208-013) 08/29/2012
Updated Version 04/08/2013
Computer Science 24 pages

Research Report

Effect of Codeword Placement on the Reliability of Erasure Coded Data Storage Systems

V. Venkatesan, I. Iliadis

IBM Research – Zurich
8803 Rüschlikon
Switzerland

Research
Almaden • Austin • Brazil • Cambridge • China • Haifa • India • Tokyo • Watson • Zurich

Effect of Codeword Placement on the Reliability of Erasure Coded Data Storage Systems

Vinodh Venkatesan and Ilias Iliadis

IBM Research – Zurich, CH-8803 Rüschlikon
{ven,ili}@zurich.ibm.com

Abstract. Modern data storage systems employ advanced erasure codes to protect data from storage node failures because of their ability to provide high data reliability at high storage efficiency. In contrast to previous studies, we consider the practical case where the length of codewords in an erasure coded system is much smaller than the number of storage nodes in the system. In this case, there exists a large number of possible ways in which different codewords can be stored across the nodes of the system. In this paper, it is shown that a declustered placement of codewords can significantly improve system reliability compared to other placement schemes. A detailed reliability analysis is presented that accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and an intelligent rebuild process that attempts to rebuild the most critical codewords first.

1 Introduction

Modern data storage systems are complex in nature consisting of several components of hardware and software. To perform a reliability analysis, we require a model that abstracts the reliability behavior of this complex system and lends itself to theoretical analysis, but at the same time, preserves the core features that affect the system failures and rebuilds. In this article, we develop and describe a relatively simple yet powerful model that captures the essential reliability behavior of an erasure coded data storage system. Using this model, we show the effect of codeword placement on the system reliability.

As an alternative to replication, storage systems employ advanced erasure codes to protect data from storage node failures because of their ability to provide high data reliability as well as high storage efficiency. The use of such erasure codes can be dated back to as early as the 1980s when they were applied in systems with redundant arrays of inexpensive disks (RAID) [1, 2]. When nodes fail, storage systems try to maintain the redundancy through node rebuild processes that use the data from the surviving nodes to reconstruct the lost data in new replacement nodes. As these rebuild processes take a finite amount of time, there exists a non-zero probability of further node failures during rebuild that can cause the system to lose enough redundant data to make some of the originally stored data irrecoverable. The average amount of time taken by the system to end up in irrecoverable data loss, also known as the mean time to

data loss, or MTTDL, is a measure of reliability commonly used for comparing different coding schemes and studying the effect of various design parameters [3]. The length of codewords in an erasure coded system is typically much smaller than the number of storage nodes in the system (e.g. RAID-6 typically uses a codeword length of 16). This implies that there exist a large number of possible ways in which codewords can be stored across the nodes of the system. However, many reliability analyses in the literature are performed under the assumption that the number of storage nodes is equal to the codeword length [1, 2, 4]. In addition, some of the reliability analyses do not account for the time taken to rebuild [4–7]. For replication-based systems, it is well-known that the MTTDL is significantly affected by the choice of placement of replicas [5, 6, 8–10]. In particular, it is known that a certain replica placement scheme, known as declustered placement, can provide significantly higher reliability than other placement schemes, especially for large storage systems [9].

This paper addresses the following practical questions regarding erasure coded systems. How does the MTTDL of a system depend on the codeword length and the number of parities in the erasure code? For a given codeword length and a given number of parities, how does the codeword placement affect the MTTDL of a system? Do the results on the effect of replica placement on MTTDL in replication-based systems extend to the effect of codeword placement on the MTTDL in erasure coded systems? How does the trade-off between storage efficiency and MTTDL depend on the codeword placement scheme?

The key contributions of this article are the following. We extend previous work in the literature by considering the general case of erasure coded systems, which includes replication-based systems. A new model enhancing previous ones is developed here to evaluate the MTTDL of erasure coded systems. The model developed captures the effect of the various system parameters as well as the effect of various codeword placement schemes. The reliability analysis is detailed, in the sense that it accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and and an intelligent rebuild process that attempts to rebuild the most critical codewords first. The validity of the model is confirmed by simulation.

The remainder of this article is organized as follows: Section 3 describes the system model considered. Section 4 describes the methodology of reliability analysis used. Using the methodology described in the previous section, Section 5 evaluates the reliability of clustered and declustered placement schemes. Section 6 provides numerical results and discusses the effect of codeword placement on reliability. Section 7 compares simulation-based MTTDL values with the theoretical predictions. Finally, the paper is concluded in Section 8.

2 Related Work

A comparison between erasure codes and replication in terms of availability in peer-to-peer systems has been presented in [11]. It has been well-established that erasure codes can provide much higher reliability than replication for the same level of storage efficiency. The trade-off, however, is in the performance as erasure codes may require Galois field arithmetic for encoding and decoding. Therefore,

Table 1. Parameters of a storage system

c	amount of data stored on each storage node (bytes)
n	number of storage nodes
$c\mu$	average read-write rebuild bandwidth of a storage node (bytes/s)
$1/\lambda$	mean time to failure of a storage node (s)
$1/\mu$	mean time to read/write c amount of data from/to a node (s)

many recent works have laid emphasis on the development of new codes as well as new encoding and decoding techniques to improve the performance of erasure coded systems (see [12] and references therein). Some works have also addressed the reliability assessment of erasure codes through simulation [13]. One thing that is common in all these works is that they essentially consider the case where the codeword length is equal to the number of nodes. In contrast, our work provides a unified framework for assessing the reliability of erasure coded systems where the codeword length may be larger than the number of nodes, in which case, there exist many possible ways of storing each codeword across the nodes in the system. This is a practically relevant case as, for performance reasons, the lengths of the erasure codes used in real storage systems are kept constant and small, whereas the number of nodes in the system grows with the system capacity. For replication-based systems, it was shown that the reliability is significantly affected by the choice of placement of replicas [9, 10, 14]. In this article, we extend these results to a more general case of maximum distance separable (MDS) erasure codes. To the best of our knowledge, this is the first work exploring the space of codeword placement for erasure codes in a homogeneous environment through both theory and simulation, which shows that codeword placement can have a significant impact on reliability.

3 System Model

The storage system is modeled as a collection of n *storage nodes* each of which stores c amount of data. In addition to the space required for the c amount of data that is stored, each node is assumed to have sufficient spare space that may be used for a distributed rebuild process (see Section 3.5) when other nodes fail. The main parameters used in the storage system model are listed in Table 1.

3.1 Storage Node

Each storage node is a fairly complex entity that comprises of disks, memory, processor, network interface, and power supply. Any of these components can fail and lead to the node either becoming temporarily unavailable, or failed. It is assumed that there is some mechanism, such as regular pinging of each node, in place to detect node failures as they occur.

Node Unavailability vs. Node Failure: The difference between node temporary unavailability and failure (or permanent unavailability) is crucial to the reliability model. Temporary node unavailability may result in temporary data unavailability. On the other hand, node failures may result in irrecoverable data loss. The primary focus of this paper is the investigation of this issue. As noted in [15], more than 90% of the node unavailabilities are transient and do not last for more than 15 minutes. As most of the unavailabilities are transient, a node

rebuild process is initiated only if a node stays unavailable for more than 15 minutes [15]. In other words, node unavailabilities lasting longer than a certain amount of time are treated as node failures.

Independence of Node Failures: Node *unavailabilities* have been observed to exhibit strong correlation that may be due to short power outages in the datacenter, or part of a rolling reboot or upgrade activity at the datacenter management layer [15]. However, there is no indication that correlations exist among node *failures*. It has been argued that disk (as opposed to node) replacement rates show correlations [16]. However, as disks have been observed to be far more reliable than other components of a node [17], the failure of a node is mainly determined by the failure of these other components. As there is no evidence that correlations exist among node failures (or permanent unavailabilities), we assume node failures to be independent in our model.

3.2 Redundancy

In erasure coded systems, the user data is divided into blocks (or symbols) of a fixed size (e.g. sector size of 512 bytes) and each set of l blocks is encoded into a set of m ($> l$) blocks, called a codeword, before storing them on m distinct nodes. In this paper, we consider (l, m)-MDS codes, in which the encoding is done, such that, any subset of l symbols of a codeword can be used to decode the l symbols of user data corresponding to that codeword. Replication-based systems, with a given replication factor r, are a subset of erasure coded systems where the parameters l and m are equal to 1 and r, respectively.

3.3 Codeword Placement

In a large storage system, the number of nodes, n, is typically much larger than the codeword length, m. Therefore, there exist many ways in which a codeword of m blocks can be stored across n nodes.

Clustered Placement: If n is divisible by m, one simple way to place codewords would be to divide the n nodes into disjoint sets, of m nodes each, and store each codeword across the nodes of a particular set. We refer to this type of data placement as *clustered* placement, and each of these disjoint sets of nodes as *clusters*. In such a placement scheme, it can be seen that no cluster stores the redundancies corresponding to the data on another cluster. The entire storage system can essentially be modeled as consisting of n/m independent clusters. Reliability behavior of a cluster under exponential failure and rebuild time distributions is well-known [1, 2, 18]. To the best of our knowledge, all prior work in the reliability analysis of erasure coded systems have solely been for clustered placement (see [12] and references therein).

Declustered Placement: A placement scheme that can potentially offer far higher reliability than the clustered placement scheme, especially as the number of nodes in the system grows, is the *declustered* placement scheme. There exist $\binom{n}{m}$ different ways of placing m symbols of each codeword across n nodes. In this scheme, all these $\binom{n}{m}$ possible ways are equally used to store all the codewords in the system. It can be seen that, in such a placement scheme, when a node fails, the redundancy corresponding to the data on the failed node is equally spread

across all the surviving nodes (as opposed to clustered placement in which it is spread only across the surviving nodes of the corresponding cluster). This allows one to use the rebuild read-write bandwidth available at all surviving nodes to do a *distributed* rebuild in parallel, which can be extremely fast when the number of nodes is large. As it turns out, this is one of the main reasons why declustered placement can offer significantly higher reliability than clustered placement for large systems.

Spread Factor: A broader set of placement schemes can be defined using the concept of *spread factor*. For each node in the system, its *redundancy spread factor* is defined as the number of nodes over which the data on that node and its corresponding redundant data are spread. In an erasure coded system, when a node fails, its spread factor determines the number of nodes which have the redundancy corresponding to the lost data, and this in turn determines the degree of parallelism that can be used in rebuilding the data lost by that node. In this paper, we will consider symmetric placement schemes in which the spread factor of each node is the same, denoted by k. Two examples of such symmetric placement schemes are the clustered and declustered placement schemes for which the spread factor, k, is equal to m and n, respectively. A number of different placement schemes can be generated by varying the spread factor, k, between m and n.

3.4 Node Failure

Based on the discussion in Section 3.1, there is no indication that node failures are correlated. Therefore, the times to node failures are modeled as independent and identically distributed random variables. Denote the cumulative distribution function of the times to node failure by F_λ, with mean, $1/\lambda$. Typically, F_λ is assumed to be exponential as this allows one to use Markov models for analysis. However, it has been observed that real-world storage devices do not have exponentially distributed failure times [16, 19]. An interesting result of this paper is that the mean time to data loss of an erasure coded storage system tends to be invariant within a large class of failure time distributions, that includes the exponential distribution and, most importantly, real-world distributions like Weibull and gamma. A similar result has been established earlier for replication-based systems [14].

3.5 Node Rebuild

When storage nodes fail, codewords lose some of their symbols and this leads to a reduction in data redundancy. The system attempts to maintain the redundancy of the system by reconstructing the lost codeword symbols using the surviving symbols of the affected codewords.

Codeword Reconstruction: For a system using an (l, m)-MDS code for redundancy, a simple way to reconstruct a codeword that has lost up to $m - l$ symbols is to read any of its l symbols, decode the original l user data blocks, re-encode these l user data blocks using the (l, m)-MDS code, and recover the lost codeword symbols. The reconstruction process takes a finite amount of time, which depends on the amount of data to read and write. Alternative methods of reconstruction based on regenerating codes have been proposed as a solution

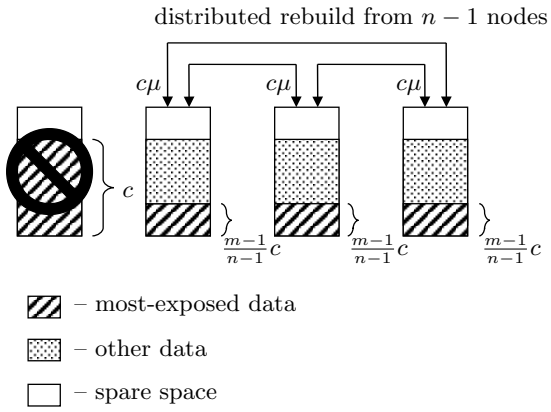


Fig. 1. Distributed rebuild in declustered placement.

to reduce the amount of data *transferred* over the storage network during reconstruction (see [20] and references therein). The effect of these methods on the system reliability is outside the scope of this paper and is a subject of further investigation.

Intelligent Rebuild: In an intelligent rebuild process, the system attempts to first recover the codewords of the user data that have the least number of codeword symbols left. These codewords are also referred to as the *most-exposed* codewords. In contrast to intelligent rebuild, one may consider a *blind* rebuild, where lost codeword symbols are being recovered in an order that is not specifically aimed at recovering the codewords with the least number of surviving symbols first. Clearly, such a blind rebuild is more vulnerable to data loss. So, in the remainder of the paper, we consider only intelligent rebuild.

Distributed Rebuild: When a storage node fails, all the codewords that had one of their symbols stored on this node are affected. For a symmetric placement scheme with spread factor k, $m \leq k \leq n$, the surviving symbols of the affected codewords are equally distributed across $k-1$ other surviving nodes of the system.

For clustered placement, i.e., $k = m$, the surviving symbols are present in the $m-1$ surviving nodes of the affected cluster. The lost symbols are recovered by reading the required codeword symbols from a set of l nodes of the corresponding surviving cluster. The lost symbols are reconstructed on the fly and directly written to a new replacement node.

For other placement schemes, i.e., $m+1 \leq k \leq n$, the surviving symbols are present in $k-1$ ($\geq m$) surviving nodes. Performing a rebuild similar to clustered placement would, in general, degrade reliability for these placement schemes. This is because, although the rebuild time would be the same (as the same amount of data is written to the new replacement node), there are more nodes ($k-1 > m-1$) that contain the surviving symbols of the affected codewords The failure of any of these nodes can result in additional symbols of the affected codewords being lost. We therefore consider instead distributed rebuild for these

placement schemes as illustrated in Fig. 1. Distributed rebuild involves reading the required codeword symbols from all the $k-1$ nodes, computing the lost codeword symbols, and writing them to the spare space of these $k-1$ nodes in such a way that no symbol is written to a node in which another symbol corresponding to the same codeword is already present. Once all lost codeword symbols are recovered, they are transferred to a new replacement node. Due to the parallel nature of distributed rebuild, the rebuild times can be extremely short for large storage systems. Such a distributed rebuild process is in fact used in practical systems [21].

Node Rebuild Bandwidth: During the rebuild process, an average read-write bandwidth of $c\mu$ bytes/s is assumed to be reserved at each node for the rebuild. This implies that the average time required to read (or write) c amount of data from (or to) a node is equal to $1/\mu$. The average rebuild bandwidth is usually only a fraction of the total bandwidth available at each node; the remainder is being used to serve user requests. Denote the cumulative distribution function of the time required to read (or write) c amount of data from (or to) a node by G_μ, and its corresponding probability density function by g_μ.

3.6 Failure and Rebuild Time Distributions

It is known that real-world storage nodes are *generally reliable*, that is, the mean time to repair a node (which is typically of the order of tens of hours) is much smaller than the mean time to failure of a node (which is typically at least of the order of thousands of hours). So, it follows that generally reliable nodes satisfy the following condition:

$$1/\mu \ll 1/\lambda, \quad \text{or} \quad \lambda/\mu \ll 1. \tag{1}$$

In the subsequent analysis, this condition implies that terms involving powers of λ/μ greater than one are negligible compared to λ/μ and can be ignored. Let the cumulative distribution functions F_λ and G_μ satisfy the following condition:

$$\mu \int_0^\infty F_\lambda(t)(1-G_\mu(t))dt \ll 1, \quad \text{with } \frac{\lambda}{\mu} \ll 1. \tag{2}$$

The results of this paper are derived for the class of failure and rebuild distributions that satisfy the above condition. In particular, the mean time to data loss of a system is shown to be insensitive to the failure distributions within this class. This result is of great importance because it turns out that this condition holds for a wide variety of failure and rebuild distributions, including, most importantly, distributions that are seen in real-world storage systems [14]. Condition (2) can also be stated in the following alternate way [14]:

$$F_\lambda(t) \ll 1 \text{ when } G_\mu(t) < 1 \text{ and } \lambda \ll \mu, \tag{3}$$

$$\mu(1-G_\mu(t)) \ll 1 \text{ when } F_\lambda(t) > 0 \text{ and } \mu \gg \lambda. \tag{4}$$

4 Reliability Analysis

The reliability analysis in this article uses a methodology similar to [9, 10, 14]. It involves a series of approximations, each of which is justified for generally reliable

nodes with failure and rebuild time distributions satisfying (2). The theoretical estimates of mean times to data loss predicted using this methodology have also been shown to match with simulations, which avoid all the approximations made in the methodology, over a wide range of system parameters [9, 10, 14].

4.1 Mean Time to Data Loss (MTTDL)

In an erasure coded system, a data loss is said to have occurred when sufficient number of blocks of at least one codeword have been lost, rendering the codeword(s) undecodeable. The average time taken for the system to end up in data loss, also referred to as the mean time to data loss, or MTTDL, is a commonly used measure that is useful for assessing trade-offs, for comparing schemes, and for estimating the effect of the various parameters on the system reliability [3].

At any point of time, the system can be thought to be in one of two modes: *fully-operational mode* or *rebuild mode*. During the fully-operational mode, all data in the system has the original amount of redundancy and there is no active rebuild process. During the rebuild mode, some data in the system has less than the original amount of redundancy and there is an active rebuild process that is trying to restore the lost redundancy. A transition from fully-operational mode to rebuild mode occurs when a node fails; we refer to this node failure that causes a transition from the fully-operational mode to the rebuild mode as a *first-node* failure. Following a first-node failure, a complex sequence of rebuilds and subsequent node failures may occur, which eventually lead the system either to irrecoverable data loss, with probability P_{DL}, or back to the original fully-operational mode by restoring all codeword symbols, with probability $1 - P_{DL}$. In other words, the probability of data loss in the rebuild mode, P_{DL}, is defined as follows:

$$P_{DL} := \Pr\left\{\begin{matrix}\text{data loss occurs before returning} \\ \text{to the fully-operational mode}\end{matrix}\middle|\ \text{system enters rebuild mode}\right\}. \quad (5)$$

Since the rebuild times are much shorter than the times to failure, when computing the time to data loss, the time spent by the system in rebuild mode can be ignored. If we ignore the rebuild times, the system timeline consists of one first-node failure after another, each of which can end up in data loss with a probability P_{DL}. It can be shown that the mean time between two successive first-node failures, converges to $1/(n\lambda)$ [14] and that the MTTDL is given by the following proposition:

Proposition 1. *Consider a system with generally reliable nodes whose failure and rebuild distributions, F_λ and G_μ, satisfy (2). Its MTTDL is given by*

$$\text{MTTDL} \approx 1/(n\lambda P_{DL}), \quad (6)$$

where P_{DL} is defined in (5). The relative error in the approximation tends to zero as λ/μ tends to zero.

Proof. See [14]. □

4.2 Probability of Data Loss in Rebuild Mode (P_{DL})

This section show how P_{DL} is estimated so that MTTDL can be obtained using Proposition 1.

Exposure Levels: Consider an erasure coded storage system with an (l, m)-MDS code. Let

$$\tilde{r} := m - l + 1. \tag{7}$$

We model the system as evolving from one exposure level to another as nodes fail and rebuilds complete. At time $t \geq 0$, let $D_j(t)$ be the amount of user data that have lost j symbols of their corresponding codewords, for $0 \leq j \leq \tilde{r}$. At time t, the system is said to be in exposure level e, $0 \leq e \leq \tilde{r}$, if $e = \max_{D_j(t)>0} j$.

Direct Path Approximation: A path to data loss following a first-node-failure event is a sequence of exposure level transitions that begins in exposure level 1 and ends in exposure level $\tilde{r}$ (data loss) without going back to exposure level 0, that is, for some $j \geq r$, a sequence of $j-1$ exposure level transitions $e_1 \to e_2 \to \cdots \to e_j$ such that $e_1 = 1$, $e_j = \tilde{r}$, $e_2, \cdots, e_{j-1} \in \{1, \cdots, \tilde{r}-1\}$, and $|e_i - e_{i-1}| = 1$, $\forall\, i = 2, \cdots, j$. To estimate P_{DL}, we need to estimate the probability of the union of *all* such paths to data loss following a first-node failure. As the set of events that can occur between exposure level 1 and exposure level $\tilde{r}$ is complex, estimating P_{DL} is a non-trivial problem. Therefore, we proceed by denoting the probability of the direct path to data loss by $P_{DL,\text{direct}}$, that is,

$$P_{DL,\text{direct}} := \Pr\{\text{exposure level path } 1 \to 2 \to \cdots \to \tilde{r}\}, \tag{8}$$

and approximate P_{DL} by $P_{DL,\text{direct}}$ using the following proposition.

Proposition 2. *Consider a system with generally reliable nodes whose failure and rebuild distributions, F_λ and G_μ, satisfy (2). Its P_{DL} is given by*

$$P_{DL} \approx P_{DL,direct}, \tag{9}$$

The relative error in the approximation tends to zero as λ/μ tends to zero.

Proof. See [9]. □

4.3 Probability of the Direct Path to Data Loss ($P_{DL,\text{direct}}$)

Consider the direct path to data loss, that is, the path $1 \to 2 \to \cdots \to \tilde{r}$ through the exposure levels. At each exposure level, the *intelligent* rebuild process attempts to rebuild the most-exposed data, that is, the data with the least number of codeword symbols left (see Section 3.5). Let the rebuild times of the most-exposed data at each exposure level in this path be denoted by R_e, $e = 1, \cdots, \tilde{r}-1$. Let t_e, $e = 2, \cdots, \tilde{r}$, be the times of transitions from exposure level $e-1$ to e following a first-node failure. Let $\tilde{n}_e$ be the number of nodes in exposure level e whose failure before the rebuild of most-exposed data causes an exposure level transition to level $e+1$. Denote the time period from t_e until the next failure of node i by $E_{t_e}^{(i)}$. The time, F_e, until the first failure among the $\tilde{n}_{e-1}$ nodes that causes the system to enter exposure level e from $e-1$, is

$$F_e := \min_{i \in \{1, \cdots, \tilde{n}_{e-1}\}} E_{t_{e-1}}^{(i)}, \quad e = 2, \cdots, \tilde{r}. \tag{10}$$

At exposure level e, let α_e be the fraction of the rebuild time R_e still left when a node failure occurs causing an exposure level transition, that is, let

$$\alpha_e := (R_e - F_{e+1})/R_e, \quad e = 1, \cdots, \tilde{r} - 2. \tag{11}$$

It can be shown that α_e is uniformly distributed in $(0, 1)$ (see Lemma 2 in Appendix A). Now, denote by $1/\mu_e$ the following conditional means of R_e:

$$1/\mu_e := E[R_e | R_{e-1}, \alpha_{e-1}], \quad e = 2, \cdots, r - 1. \tag{12}$$

The actual values of $1/\mu_e$ depend on the codeword placement and this will be further discussed in later sections of this paper. Now, the distribution of R_e given R_{e-1} and α_{e-1} could be modeled in several ways. We consider the model B presented in [14], namely,

$$R_e | R_{e-1}, \alpha_{e-1} = 1/\mu_e \qquad w.p.\, 1 \text{ for } e = 2, \cdots, \tilde{r} - 1. \tag{13}$$

This model assumes that the rebuild time R_e is determined completely by R_{e-1} and α_{e-1} and no new randomness is introduced in the rebuild time of exposure level e. For further discussion on this model see [14]. Under this model, the probability of the direct path to data loss is given by the following proposition.

Proposition 3. *Consider a system with generally reliable nodes whose failure and rebuild distributions, F_λ and G_μ, satisfy (2). Consider the direct path $1 \to 2 \to \cdots \to \tilde{r}$ through the exposure levels in which the rebuild times R_e satisfy (13). The probability of this direct path is given by*

$$P_{DL,direct} \approx \lambda^{\tilde{r}-1} \times \tilde{n}_1 \cdots \tilde{n}_{\tilde{r}-1} \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_{\tilde{r}-1}=0}^{\infty} \int_{a_1=0}^{1} \cdots \int_{a_{\tilde{r}-2}=0}^{1} \Bigg(\tau_1 \cdots \tau_{\tilde{r}-1} g_{\mu_1}(\tau_1)$$
$$\times \delta\left(\tau_2 - \frac{1}{\mu_2}\right) \cdots \delta\left(\tau_{\tilde{r}-1} - \frac{1}{\mu_{\tilde{r}-1}}\right) da_{\tilde{r}-2} \cdots da_1 d\tau_{\tilde{r}-1} \cdots d\tau_1 \Bigg). \tag{14}$$

The relative error in the approximation tends to zero as λ/μ tends to zero.

Proof. See Appendix A. □

5 Effect of Codeword Placement on Reliability

In this section, we consider different codeword placement schemes as discussed in Section 3.3. We wish to estimate their reliability in terms of their MTTDL and understand how codeword placement affects data reliability. To use the expression (14) for $P_{DL,\text{direct}}$, we need to compute the conditional means of rebuild times in each exposure level, $1/\mu_e$, $e = 1, \cdots, \tilde{r} - 1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, \tilde{r} - 1$. The values of these quantities depend on the underlying codeword placement and the nature of the rebuild process used. Now, denote the kth raw moment of the rebuild distribution G_μ by $M_k(G_\mu)$. The MTTDL of clustered and declustered codeword placement schemes are given by the following propositions.

Proposition 4. *Consider a storage system using clustered codeword placement with generally reliable nodes whose failure and rebuild distributions satisfy (2). Its mean time to data loss is given by*

$$\text{MTTDL}^{clus.} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{\binom{m-1}{l-1}} \frac{M_1^{m-l}(G_\mu)}{M_{m-l}(G_\mu)}. \tag{15}$$

The relative error in the above approximation tends to zero as λ/μ tends to zero.

Proof. See Appendix B. □

Proposition 5. *Consider a storage system using declustered codeword placement with generally reliable nodes whose failure and rebuild distributions satisfy (2). Its mean time to data loss is given by*

$$\text{MTTDL}^{declus.} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{(m-l)!}{(l+1)^{m-l}} \frac{M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)}{M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)} \prod_{e=1}^{m-l-1} \left(\frac{n-e}{m-e}\right)^{m-l-e}. \tag{16}$$

The relative error in the above approximation tends to zero as λ/μ tends to zero.

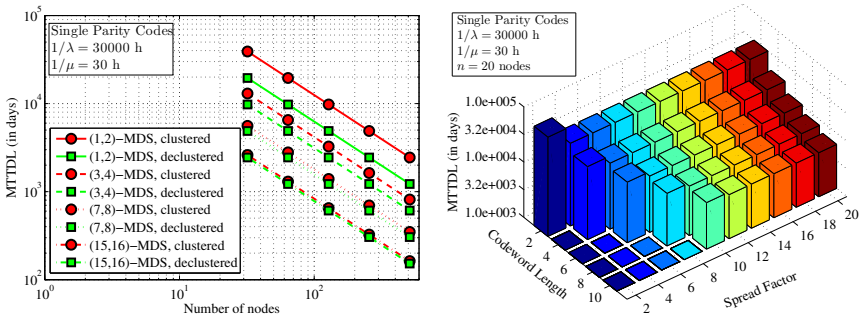
Proof. See Appendix C. □

Remark 1. The expressions for MTTDL obtained in this paper are better approximations for smaller values of λ/μ. This implies that, if simulation-based MTTDL values match the theoretically predicted MTTDL values for a certain value of λ/μ, it will also match for all smaller values of λ/μ. This fact is used in Section 7, where simulations are shown to match theory for values of λ/μ that are much larger than those observed in real-world storage systems, thereby establishing the applicability of the theoretical results to real-world storage systems.

6 Numerical Results

In this section, we compare the MTTDLs of (l, m)-MDS code based systems for clustered and declustered placement schemes for various choice of parameters l and m with the help of figures.

Single Parity Codes: Single parity (l, m)-MDS codes correspond to the case where $m-l = 1$. When $l = 1$, this corresponds to two-way replication. For higher values of l, this corresponds to RAID-5 [1]. It is observed that the MTTDL of single parity codes under both placement schemes are directly proportional to the square of the mean time to node failure, $1/\lambda$, and inversely proportional to the mean time to read all contents of a node during rebuild, $1/\mu$. In addition, the MTTDL values are seen to be independent of the underlying rebuild distribution. The result for clustered placement is well known since the 1980s when the reliability of RAID-5 systems were studied [1]. Fig. 2(a) illustrates the MTTDL behavior of single parity codes with respect to the number of nodes in the system. It is seen that the MTTDL is inversely proportional to the number of nodes for both clustered and declustered placement schemes.



(a) MTTDL vs number of nodes n

(b) MTTDL vs codeword length m and spread factor k for $n = 20$

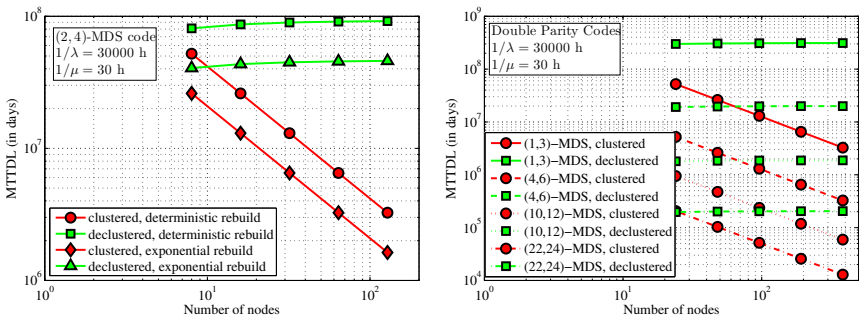
Fig. 2. MTTDL for single parity codes with $1/\lambda = 30000$ h and $1/\mu = 30$ h

Fig. 2(b) shows how the MTTDL varies as a function of both the codeword length m and the spread factor k for single parity codes, for a given number of nodes n. In Fig. 2(b), clustered placement corresponds to the cases where the spread factor is equal to the codeword length, and declustered placement corresponds to the case where the spread factor is equal to the number of nodes. It is observed that the clustered placement scheme has slightly higher MTTDL values than other placement schemes, and that increasing the codeword length decreases the MTTDL.

Double Parity Codes: It is observed that the MTTDL of double parity codes under both placement schemes are directly proportional to the cube of the mean time to node failure, $1/\lambda$, and inversely proportional to the square of the mean time to read all contents of a node during rebuild, $1/\mu$. The result for clustered placement is well known in the context of RAID-6 systems [2].

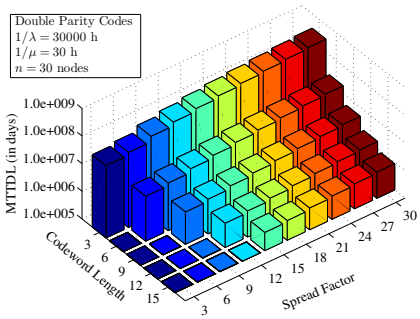
In contrast to single parity codes, it is seen that the MTTDL of double parity codes depends on rebuild distribution. For deterministic rebuild times, the ratios $M_1^2(G_\mu)/M_2(G_\mu)$ and $M_1^2\left(G_{\frac{n-1}{m-1}\mu}\right)/M_2\left(G_{\frac{n-1}{m-1}\mu}\right)$ become one. However, for random rebuild times, these ratios are upper-bounded by one by Jensen's inequality. The MTTDL of a system using a $(2,4)$-MDS code is plotted against the number of nodes in the system for clustered and declustered placements, as well as for deterministic and exponential rebuild times, in Fig. 3(a). It is observed that the rebuild time distribution scales down the MTTDL, but leaves the behavior with respect to the number of nodes, n, unaffected.

In contrast to single parity codes, the difference in MTTDL between the two schemes can be significant, depending on the number of nodes, n, in the system. This is because, the MTTDL of clustered placement is inversely proportional to n, whereas the MTTDL of declustered placement is roughly invariant with respect to n. This is illustrated in Fig. 3(b) in which MTTDL of double parity codes is plotted against the number of nodes, n, in a log-log scale. The lines corresponding to clustered placement have a slope of -1, whereas the lines corresponding to declustered placement have a slope of roughly 0. It is also ob-



(a) MTTDL vs number of nodes n for a (2, 4)-MDS code illustrating the effect of various rebuild distribution

(b) MTTDL vs number of nodes n for double parity codes



(c) MTTDL vs codeword length m and spread factor k for $n = 30$

Fig. 3. MTTDL for double parity codes with $1/\lambda = 30000$ h and $1/\mu = 30$ h

served from Fig. 3(b) that longer codes, which are more desirable as they have higher storage efficiency, can have better MTTDL with declustered placement than shorter codes with clustered placement for large systems. This is seen, for example, by observing the lines corresponding to $(4, 6)$-MDS code with declustered placement and $(1, 3)$-MDS code with clustered placement, for $n > 100$. Just like in the case of single parity codes, the difference in MTTDL between clustered and declustered is observed to be smaller for larger values of the codeword length, m. Fig. 3(c) shows how the MTTDL varies as a function of both the codeword length m and the spread factor k for double parity codes, for a given number of nodes, n.

Codes with Higher Number of Parities: Comparing the MTTDL values of clustered placement in (15) with those of declustered placement in (16), we observe that they are both directly proportional to the $(m-l+1)$th power of the mean time to node failure $1/\lambda$, and inversely proportional to the $(m-l)$ power of the mean time to node rebuild $1/\mu$. This is a general trend in the MTTDL behavior of data storage systems. However, in contrast to clustered placement, which always scales inversely proportional to the number of nodes, the MTTDL of declustered placement is observed to scale differently with the number of nodes for different values of $\tilde{r} = m - l + 1$. In particular, for codes with more than

two parities, the MTTDL of declustered placement increases with n. This shows that, by changing the codeword placement scheme, one can influence the scaling of MTTDL with respect to the number of nodes n, resulting in a tremendous improvement in reliability for large storage systems.

7 Simulations

Event-driven simulations are used to verify the theoretical estimates of MTTDL of erasure coded systems for two placement schemes, namely, clustered and declustered. The simulations are more involved than the theoretical analysis as they do not make any of the approximations made in theory. Despite this fact, it is found that the theoretical estimates match the simulation results for a wide range of parameters, including the parameters generally observed in practice, thereby validating the applicability of the reliability analysis to real-world storage systems.

7.1 Simulation Method

The storage system is simulated using an event-driven simulation with three types of events that drive the simulation time forward: *(a) failure events*, *(b) rebuild-complete events*, and *(c) node-restore events*. The state of the system is maintained by the following variables: `time`, the simulated time; `nActiveNodes`, the number of active (surviving) nodes in the system; `failTimes`, the times of next failure of each active node generated according to the distribution F_λ; `failedNodes`, the indices of all failed nodes; `exposureLevel`, the exposure level; and a vector of length $(r+1)$, `dataExposure` $= (D_0, \cdots, D_{\tilde{r}})$, where D_e is the number of codewords that have lost e symbols, $e = 1, \cdots, \tilde{r}$. The values of these variables are updated at each event, and when $D_{\tilde{r}} > 0$, data loss is said to have occurred and the simulation ends. For each set of parameters, the simulation is run 100 times, and the MTTDL and its 95% confidence intervals are computed. To obtain the time to data loss for declustered placement, the simulation is run for all n nodes. In contrast, for clustered placement, n/m simulations are run for one cluster, that is, m nodes, and the time to data loss of the whole system is obtained by taking the minimum of the times to data loss obtained in each of the n/m simulations. This is because the clusters are independent of each other and the number of clusters is n/m.

Failure Event Besides updating `time`, a failure event triggers the following: (i) decreasing `activeNodes` by one and increasing `exposureLevel` by one (recall that, for the declustered scheme, any node failure causes an exposure level transition, and that, for the clustered scheme, only one cluster is being simulated and therefore any node failure in that cluster causes an exposure level transition), (ii) scheduling the next failure event based on `failTimes`, (iii) updating `dataExposure` by taking partial rebuild of the most exposed data into account, and (iv) scheduling the rebuild-complete event based on the most exposed data in `dataExposure`, the placement scheme used (which determines the parallelism

that can be exploited and therefore the speed of rebuild), network rebuild bandwidth limitations, and the rebuild distribution. By the nature of the rebuild process, data placement is preserved, that is, declustered remains declustered and clustered remains clustered. This is because, when the placement is declustered, critical blocks are read from and written to all nodes at the same time and the new codeword symbols are placed such that declustering is preserved. When the placement is clustered, the codeword symbols are created in a new node directly, which again preserves the placement. One main difference between declustered and clustered placement is how the data exposure vector changes at each exposure level transition. It was observed in the previous sections that the main reason for declustered placement to have a higher reliability is the fact that the amount of most-exposed data at each exposure level decreases significantly as the system enters higher exposure levels. Therefore, proper computation of data exposure vector at each exposure level transition for declustered placement is an important step in its reliability simulation. Whereas the computation of data exposure vector for clustered placement is fairly straightforward, the computation of data exposure vector for declustered placement is more involved.

Data Exposure Vector for Declustered Placement For declustered placement at exposure level e, when a failure occurs, the data exposure vector, `dataExposure`, is updated from $(D_0, D_1, \cdots, D_e, 0, \cdots, 0)$ to $(D'_0, D'_1, \cdots, D'_e, D'_{e+1}, 0, \cdots, 0)$ as follows. Let $\tilde{n}$ denote the number of active nodes in the system at exposure level e. For $j = 0, \cdots, e-1$, the number of codewords that have $m-j$ surviving symbols in exposure level e is equal to D_j. These $m-j$ symbols are equally spread across the $\tilde{n}$ surviving nodes of the system due to the nature of declustered placement and distributed rebuild. Therefore, when an additional node failure occurs, $\frac{m-j}{\tilde{n}} D_j$ codewords lose their $(j+1)$th symbols, for $j = 0, \cdots, e-1$. So, D'_j, $j = 0, \cdots, e-2$ is given by

$$D'_0 = D_0 - \frac{m}{\tilde{n}} D_0, \tag{17}$$

$$D'_j = D_j - \frac{m-j}{\tilde{n}} D_j + \frac{m-j+1}{\tilde{n}} D_{j-1}, \quad \text{for } j = 1, \cdots, e-2, \tag{18}$$

If α denotes the fraction of rebuild time at exposure level e still left when a transition to exposure level $e+1$ occurred, then D'_{e+1} is given by:

$$D'_{e+1} = \frac{m-e}{\tilde{n}} \alpha D_e. \tag{19}$$

This is because, $\frac{m-e}{\tilde{n}} \alpha D_e$ codewords lose their $(e+1)$th symbol during the exposure level transition. However, an additional symbol was created by the rebuild process in exposure level e for each of the $(1-\alpha)D_e$ most-exposed codewords. Therefore,

$$D'_{e-1} = D_{e-1} - \frac{m-e+1}{\tilde{n}} D_{e-1} + \frac{m-e+2}{\tilde{n}} D_{e-2} + (1-\alpha) D_e, \tag{20}$$

Moreover, $\frac{m-e+1}{\tilde{n}} D_{e-1}$ codewords lose their eth symbol during the exposure level transition. Therefore, it follows that D'_e is given by

$$D'_e = D_e - \frac{m-e}{\tilde{n}} \alpha D_e + \frac{m-e+1}{\tilde{n}} D_{e-1} - (1-\alpha) D_e. \tag{21}$$

Data loss occurs when $D_{\tilde{r}}$ becomes positive.

Rebuild-Complete Event A rebuild-complete event updates `time` and triggers the following: (i) decreasing `exposureLevel` by one, (ii) at exposure level e, $e = 1, \cdots, r-1$, updating `dataExposure` by adding D_e to D_{e-1} and setting D_e to zero (this means that the rebuild process always creates symbols of the most exposed data first, or in other words, an intelligent rebuild is done), (iii) scheduling the next rebuild-complete event based on the most exposed codewords, the placement scheme, and the rebuild distribution. Besides these, there are a few other updates that differ based on placement: for declustered placement, when all codewords have m symbols, that is, when the exposure level becomes 0, a node-restore event is scheduled. A node-restore event occurs at the time when all the newly restored codeword symbols have been successfully transferred to new replacement nodes and the number of nodes is brought back to n. The number of nodes to restore is stored in `nodesToRestore`. For clustered placement, `activeNodes` is increased by one (because codeword symbols are being directly created in a new node and so a node-restore event is not required), and a failure time for the newly restored node is generated using the failure distribution F_λ.

Node-Restore Event From the preceding, it follows that a node-restore event needs to be scheduled only for declustered placement. Besides updating the simulated time, this event increases `activeNodes` by `nodesToRestore` and sets `nodesToRestore` to zero. Failure times for the newly restored nodes are scheduled using the failure distribution F_λ.

7.2 Theory vs. Simulation

Although some of the assumptions used in the theoretical analysis, such as independence of node failures, are also used in the simulation, the simulation results reflect a more realistic picture of the systems's reliability. This is because of the following key differences between the theoretical analysis and the simulations. The theoretical estimate of MTTDL in (6) takes into account only the time spent by the system in the fully-operational mode and ignores the time spent in rebuild mode, whereas the simulations do not ignore the rebuild times when calculating the times to data loss. Furthermore, in (9), P_{DL} is approximated by the probability of the direct path to data loss. In simulations however, all the complex trajectories of the system through the different exposure levels are simulated by simulating random node failure events and updating the data exposure vector by taking partial rebuilds into account. In the theoretical analysis, the time required to restore new nodes in a declustered placement scheme (following

Table 2. Range of values of different simulation parameters

Parameter	Meaning	Range
c	amount of data stored on each node	12 TB
n	number of storage nodes	4 to 200
$m-l$	number of parities	$1, 2$
b	average rebuild bandwidth at each storage node	96 MB/s
$1/\lambda$	mean time to failure of a node	1000 h to 10000 h
$1/\mu$	average time to read/write c amount of data from/to a node during rebuild ($1/\mu = c/b$)	35 h

successful rebuild of lost codeword symbols in the spare space of surviving nodes) is ignored, whereas in the simulations, the time to restore new nodes is simulated as well. In addition, other approximations made in the analysis, such as neglecting the effect of the transient period of the system, are implicitly avoided in the simulations. Therefore, the simulations reflect a more comprehensive picture of the system behavior than what is assumed in theory.

7.3 Simulation Results

Table 2 shows the range of parameters used for the simulations. Typical values for practical systems are used for all parameters, except for the mean times to failure of a node, which have been chosen artificially low (1000 h to 10000 h) to run the simulations fast. The running times of simulations with practical values of the mean times to node failure, which are of the order of 10000 h or higher, are prohibitively high; this is due to the fact that P_{DL} becomes extremely low thereby making the number of first-node-failure events that need to be simulated (along with the other complex set of events that restore all lost codeword symbols following each first-node-failure event) extremely high for each run of the simulation. It is seen that, despite the unrealistically low values of mean times to node failure, the simulation-based values are a good match to the theoretical estimates. This observation in conjunction with Remark 1 implies that the theoretical estimates will also be accurate for realistic values of mean times to node failure, $1/\lambda$, which are generally much higher.

Figure 4 shows the comparison between the theoretically-predicted MTTDL values and the simulation-based MTTDL values for systems using $(3, 4)$ and $(6, 8)$ MDS codes. The simulation-based MTTDLs are observed to be in agreement with the theoretical predictions.

8 Conclusion

The reliability of erasure coded systems was studied with a detailed analytical model that accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and the fact that modern systems utilize an intelligent rebuild process that rebuilds the most critical codewords first. It was shown that the mean time to data loss of erasure coded systems are practically insensitive to distribution of times to node failure but sensitive to the distribution of node rebuild times. In particular, it was shown

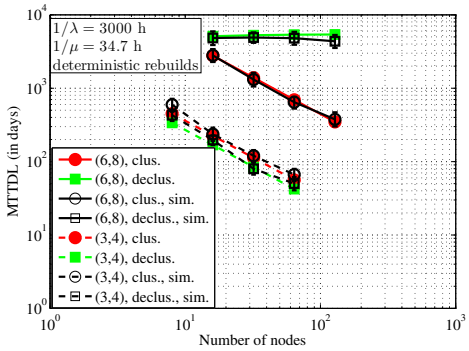


Fig. 4. MTTDL of two different erasure codes with the same storage efficiency for a system with mean time to node failure $1/\lambda = 3000$ h and mean time to read all contents of a node during rebuild $1/\mu = 34.7$ h.

that random rebuild times result in lower MTTDL values compared to deterministic rebuild times. The codeword placement scheme, and the rebuild process used, are major factors that influence the scaling of MTTDL with the number of nodes in the system. Declustered codeword placement with distributed rebuild was shown to potentially have significantly larger values of MTTDL compared to clustered codeword placement as the number of nodes in the system increases. Simulations were used to confirm the validity of the theoretical model. Extensions of this work to non-MDS codes and correlated failures are subjects of further investigation.

References

1. D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. 1988 ACM SIGMOD International Conference on Management of Data*, 1988, pp. 109–116.
2. P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: high-performance, reliable secondary storage," *ACM Computing Surveys*, vol. 26, no. 2, pp. 145–185, June 1994.
3. A. Thomasian and M. Blaum, "Higher reliability redundant disk arrays: Organization, operation, and coding," *ACM Trans. Storage*, vol. 5, no. 3, pp. 1–59, 2009.
4. D. Leong, A. G. Dimakis, and T. Ho, "Distributed storage allocation for high reliability," in *Proc. IEEE International Conference on Communications*, 2010, pp. 1–6.
5. M. Leslie, J. Davies, and T. Huffman, "A comparison of replication strategies for reliable decentralised storage," *Journal of Networks*, vol. 1, no. 6, pp. 36–44, December 2006.
6. A. Thomasian and M. Blaum, "Mirrored disk organization reliability analysis," *IEEE Transactions on Computers*, vol. 55, pp. 1640–1644, December 2006.
7. X. Li, M. Lillibridge, and M. Uysal, "Reliability analysis of deduplicated and erasure-coded storage," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 4–9, January 2011.
8. Q. Xin, E. L. Miller, and T. J. E. Schwarz, "Evaluation of distributed recovery in large-scale storage systems," in *Proc. 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04)*, 2004, pp. 172–181.

9. V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, "Reliability of clustered vs. declustered replica placement in data storage systems," in *Proc. 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'11)*, 2011, pp. 307–317.
10. V. Venkatesan, I. Iliadis, and R. Haas, "Reliability of data storage systems under network rebuild bandwidth constraints," in *Proc. 20th Annual IEEE Int'l Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'12)*, 2012, pp. 189 –197.
11. H. Weatherspoon and J. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in *Proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, Mar. 2002, pp. 328–338.
12. J. S. Plank and C. Huang, "Tutorial: Erasure coding for storage applications," Slides presented at 11th Usenix Conference on File and Storage Technologies (FAST'13), San Jose, February 2013.
13. K. M. Greenan, E. L. Miller, and J. Wylie, "Reliability of flat XOR-based erasure codes on heterogeneous devices," in *Proc. 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'08)*, June 2008, pp. 147–156.
14. V. Venkatesan and I. Iliadis, "A general reliability model for data storage systems," in *Proc. 9th Int'l Conference on Quantitative Evaluation of Systems (QEST'12)*, 2012, pp. 209–219.
15. D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *Proc. 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)*, 2010, pp. 61–74.
16. B. Schroeder and G. A. Gibson, "Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you?" *ACM Transactions on Storage*, vol. 3, no. 3, pp. 1–31, October 2007.
17. W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky, "Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics," *ACM Transactions on Storage*, vol. 4, no. 3, pp. 1–25, November 2008.
18. S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, 2006, pp. 1–9.
19. E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *Proc. 5th USENIX conference on File and Storage Technologies (FAST'07)*, 2007, pp. 17–28.
20. A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network coding for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, 2011.
21. IBM, "XiV Storage System Specifications." [Online]. Available: www.xivstorage.com

Appendix

A Proof of Proposition 3

Consider a sample direct path with $R_e = \tau_e$, $e = 1, \cdots, \tilde{r} - 1$, and $\alpha_e = a_e$, $e = 1, \cdots, \tilde{r} - 2$.[1] Denote the vector $(\tau_1, \cdots, \tau_{\tilde{r}-1})$ by $\boldsymbol{\tau}$ and $(a_1, \cdots, a_{\tilde{r}-2})$ by

[1] More strictly, we consider a direct path to data loss with $\tau_e < R_e \leq \tau_e + \delta\tau_e$, $e = 1, \cdots, \tilde{r} - 1$, and $a_e < \alpha_e \leq \delta a_e$, $e = 1, \cdots, \tilde{r} - 2$, where $\delta\tau_e$ and δa_e are positive infinitesimal quantities, but we leave this out for notational convenience.

$\boldsymbol{a}$ for notational convenience. Then, the probability of this direct path, denoted by $P_{DL,\text{direct}}(\boldsymbol{\tau}, \boldsymbol{a})$, is

$$\begin{aligned} P_{DL,\text{direct}}(\boldsymbol{\tau}, \boldsymbol{a}) &= \Pr\{R_1 = \tau_1\} \times \Pr\{F_2 < R_1 | R_1 = \tau_1\} \\ &\quad \times \Pr\{\alpha_1 = a_1 | R_1 = \tau_1, F_2 < R_1\} \\ &\quad \times \Pr\{R_2 = \tau_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1\} \\ &\quad \times \Pr\{F_3 < R_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1, R_2 = \tau_2\} \\ &\quad \cdots \times \Pr\big\{F_{\tilde{r}} < R_{\tilde{r}-1} \big| R_e = \tau_e, F_{e'+1} < R_{e'}, \alpha_{e'} = a_{e'}, \\ &\qquad \forall\, e \in \{1, \cdots, \tilde{r}-1\}, \forall\, e' \in \{1, \cdots, \tilde{r}-2\}\big\}. \end{aligned} \tag{22}$$

If we denote the mean of R_1 by $1/\mu_1$, based on the rebuild model described in Section 3.5, it follows that R_1 is distributed according to some distribution G_{μ_1} that satisfies (2), that is, $R_1 \sim G_{\mu_1}$. Therefore, the first term in (22) reduces to

$$\Pr\{R_1 = \tau_1\} = g_{\mu_1}(\tau_1)\delta\tau_1, \tag{23}$$

where $\delta\tau_1$ denotes an infinitesimal increment in τ_1. The remaining terms in (22) fall into three types:

$$\begin{aligned} &\text{A: } \Pr\big\{F_e < R_{e-1} \big| R_{e'} = \tau_{e'}, F_{e''+1} < R_{e''}, \alpha_{e''} = a_{e''}, \\ &\qquad \forall\, e' \in \{1, \cdots, e-1\}, \forall\, e'' \in \{1, \cdots, e-2\}\big\}, \end{aligned} \tag{24}$$

$$\begin{aligned} &\text{B: } \Pr\big\{\alpha_e = a_e \big| R_{e'} = \tau_{e'}, F_{e'+1} < R_{e'}, \alpha_{e''} = a_{e''}, \\ &\qquad \forall\, e' \in \{1, \cdots, e\}, \forall\, e'' \in \{1, \cdots, e-1\}\big\}, \end{aligned} \tag{25}$$

$$\begin{aligned} &\text{C: } \Pr\big\{R_e = \tau_e \big| R_{e'} = \tau_{e'}, F_{e'+1} < R_{e'}, \alpha_{e'} = a_{e'}, \\ &\qquad \forall\, e' \in \{1, \cdots, e-1\}\big\}. \end{aligned} \tag{26}$$

The following three lemmas give the expressions for terms of type A, B, and C.

Lemma 1. *Expressions of type A given by (24) reduce to*

$$\Pr\{F_e < R_{e-1} | R_{e-1} = \tau_{e-1}\} \approx \tilde{n}_{e-1}\lambda\tau_{e-1}, \tag{27}$$

for $e = 2, \cdots, \tilde{r}$, where the approximation holds for systems with generally reliable nodes satisfying (2). The relative error in the approximation tends to zero as λ/μ tends to zero.

Proof. See Appendix D. □

Lemma 2. *Expressions of type B given by (25) reduce to*

$$\Pr\{\alpha_e = a_e | R_e = \tau_e, F_{e+1} < R_e\} \approx \delta a_e, \tag{28}$$

for $e = 1, \cdots, \tilde{r}-2$, where the approximation holds for systems with generally reliable nodes satisfying (2). Here, δa_e denotes an infinitesimal increment of a_e. The relative error in the approximation tends to zero as λ/μ tends to zero.

Proof. See Appendix E. □

Lemma 3. *For a direct sample path $1 \to 2 \to \cdots \to \tilde{r}$ through the exposure levels in which the rebuild times R_e satisfy (13), expressions of type C given by (26) reduce to*

$$\Pr\{R_e = \tau_e | R_{e-1} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}\} = \delta(\tau_e - 1/\mu_e)\delta\tau_e \tag{29}$$

for $e = 2, \cdots, \tilde{r} - 1$. Here, $\delta(\tau_e - 1/\mu_e)$ denotes the Dirac delta function with a spike at $1/\mu_e$, and $\delta\tau_e$ denotes an infinitesimal increment of τ_e.

Proof. See Appendix F. □

Substituting (23), (27), (28), and (29) in (22), we obtain

$$\begin{aligned} P_{DL,\text{direct}}(\boldsymbol{\tau}, \boldsymbol{a}) \approx \lambda^{\tilde{r}-1} \times \tilde{n}_1 \cdots \tilde{n}_{\tilde{r}-1} \times \tau_1 \cdots \tau_{\tilde{r}-1} \times g_{\mu_1}(\tau_1) \times \delta a_1 \cdots \delta a_{\tilde{r}-2} \\ \times \delta\tau_1 \cdots \delta\tau_{\tilde{r}-1} \times \delta(\tau_2 - 1/\mu_2) \cdots \delta(\tau_{\tilde{r}-1} - 1/\mu_{\tilde{r}-1}). \end{aligned} \tag{30}$$

The probability of the direct path to data loss, $P_{DL,\text{direct}}$, is the summation of the probabilities, $P_{DL,\text{direct}}(\boldsymbol{\tau}, \boldsymbol{a})$, of all possible sample direct paths. As the infinitesimal increments in (30) tend to zero, the summation becomes an integral resulting in (14). □

B Proof of Proposition 4

The rebuild process in clustered placement always involves reading data from l nodes of the affected cluster at an average bandwidth of $c\mu$ from each node, computing the lost codeword symbols on-the-fly, and writing them to a spare node at an average bandwidth of $c\mu$. Therefore, in exposure level 1, the average time to rebuild the c amount of lost data is given by

$$1/\mu_1^{\text{clus.}} = 1/\mu. \tag{31}$$

In the direct path approach to data loss in a system using clustered placement, we need to consider only successive failures of nodes belonging to the same cluster. As no cluster shares the redundancies corresponding to the data on another cluster, the number of nodes, $\tilde{n}_e$, whose failure can cause a transition to the next exposure level is equal to the number of surviving nodes in the affected cluster in exposure level e, that is,

$$\tilde{n}_e^{\text{clus.}} = m - e. \tag{32}$$

When the system enters exposure level e, all of the most-exposed codewords that were unrebuilt in exposure level $e-1$ lose their eth codeword symbol. Therefore, given that the rebuild time in the previous exposure level was $R_{e-1} = \tau_{e-1}$, and the fraction $\alpha_{e-1} = a_{e-1}$, the conditional mean $1/\mu_e^{\text{clus.}}$ is given by

$$1/\mu_e^{\text{clus.}} = a_{e-1}\tau_{e-1}, \quad e = 2, \cdots, \tilde{r} - 1. \tag{33}$$

By substituting the values of $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ from (31), (32), and (33), into (14), successively evaluating the integrals in (14), we get

$$P_{DL}^{\text{clus.}} \approx \frac{\lambda^{m-l}}{\mu^{m-l}} \binom{m-1}{l-1} \frac{M_{m-l}(G_\mu)}{M_1^{m-l}(G_\mu)}. \tag{34}$$

Substituting (34) into (6), we obtain (15). □

C Proof of Proposition 5

The distributed rebuild process in each exposure level e involves reading the required codeword symbols of the data to be rebuilt from all the $n-e$ surviving nodes of the system, computing the lost codeword symbols, and writing them to the spare space of these nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. In exposure level 1, this process requires reading lc amount of data, as well as writing c amount of data, from and to all $n-e$ surviving nodes in parallel. As each of the $n-e$ nodes has an average read-write rebuild bandwidth of $c\mu$, the average time to rebuild the c amount of lost data is given by

$$1/\mu_1^{\text{declus.}} = (l+1)/((n-1)\mu). \tag{35}$$

Due to the nature of declustered placement, the failure of any of the surviving $n-e$ nodes at exposure level e before rebuild will cause a transition to exposure level $e+1$. Therefore,

$$\tilde{n}_e^{\text{declus.}} = n-e. \tag{36}$$

When the system enters exposure level e, in contrast to clustered placement, *not* all the most-exposed codewords whose symbols were unrebuilt in exposure level $e-1$ lose their eth codeword symbol. Due to the nature of declustered codeword placement, the newly failed node stored codeword symbols corresponding to only a fraction $(m-e+1)/(n-e+1)$ of these most-exposed codewords. Furthermore, as the rebuild in exposure level e involves only $n-e$ nodes (versus $n-e+1$ nodes in exposure level $e-1$) the speeds of rebuild in exposure levels e and $e-1$ differ by a factor $(n-e+1)/(n-e)$. Taking these effects into account, and given that the rebuild time in the previous exposure level was $R_{e-1} = \tau_{e-1}$, and the fraction $\alpha_{e-1} = a_{e-1}$, the conditional mean $1/\mu_e^{\text{declus.}}$ is given by

$$1/\mu_e^{\text{clus.}} = \frac{m-e+1}{n-e} a_{e-1}\tau_{e-1}, \quad e = 2, \cdots, \tilde{r}-1. \tag{37}$$

By substituting the values of $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ from (35), (36), and (37), into (14), successively evaluating the integrals in (14), and substituting the result in (6), we obtain (16). □

D Proof of Lemma 1

Expressions of the form (24) denote the conditional probability of transition from exposure level $e-1$ to e. Given that the rebuild time $R_{e-1} = \tau_{e-1}$, the event $F_e < R_{e-1}$ is independent of all other conditioning terms in (24). Removing these other conditioning terms, (24) becomes

$$\Pr\{F_e < R_{e-1} | R_{e-1} = \tau_{e-1}\} = \Pr\{F_e < \tau_{e-1}\}. \tag{38}$$

Here (38) follows from the fact that the time to next node failure, F_e, and the time to rebuild the most-exposed data, R_{e-1}, are independent. Substituting for F_e from (10), we have

$$\Pr\{F_e < \tau_{e-1}\} = \Pr\left\{\min_{i\in\{1,\cdots,\tilde{n}_{e-1}\}} E^{(i)}_{t_{e-1}} < \tau_{e-1}\right\} \tag{39}$$

$$= 1 - \Pr\left\{\min_{i\in\{1,\cdots,\tilde{n}_{e-1}\}} E^{(i)}_{t_{e-1}} \geq \tau_{e-1}\right\} \tag{40}$$

$$= 1 - \Pr\left\{E^{(i)}_{t_{e-1}} \geq \tau_{e-1}\, \forall\, i \in \{1,\cdots,\tilde{n}_{e-1}\}\right\} \tag{41}$$

$$= 1 - \left(\Pr\left\{E^{(1)}_{t_{e-1}} \geq \tau_{e-1}\right\}\right)^{\tilde{n}_{e-1}} \tag{42}$$

$$= 1 - \left(1 - \Pr\left\{E^{(1)}_{t_{e-1}} < \tau_{e-1}\right\}\right)^{\tilde{n}_{e-1}}. \tag{43}$$

Here, (42) follows from the fact that $E^{(i)}_{t_{e-1}}$ are independent random variables. It is known that, during the stationary period of the system, $E^{(i)}_{t_e}$ are distributed according to $\widetilde{F}_\lambda$ given by [14, Lemma 2]

$$\widetilde{F}_\lambda(t) := \lambda \int_0^t (1 - F_\lambda(\tau))d\tau. \tag{44}$$

Therefore,

$$\Pr\left\{E^{(1)}_{t_{e-1}} < \tau_{e-1}\right\} = \widetilde{F}_\lambda(\tau_{e-1}) = \lambda \int_0^{\tau_{e-1}} (1 - F_\lambda(\tau))d\tau \tag{45}$$

$$= \lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right). \tag{46}$$

Here, (46) follows from (3). Substituting (46) in (43), we get

$$\Pr\{F_e < \tau_{e-1}\} = 1 - \left(1 - \lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right)\right)^{\tilde{n}_{e-1}} \tag{47}$$

$$= \tilde{n}_{e-1}\lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right) \approx \tilde{n}_{e-1}\lambda\tau_{e-1}, \tag{48}$$

where the approximation (48) holds good for systems with generally reliable nodes satisfying (2). From (38) and (48), we observe that the type A expressions of the form (24) can be reduced to (27). □

E Proof of Lemma 2

Type B terms of the form (25) denote the conditional probability that the fraction, α_e, of rebuild time, R_e, still left when an exposure level transition from e to $e+1$ occurred, is equal to a_e. Given that $R_e = \tau_e$ and $F_{e+1} < R_e$, the fraction α_e is independent of the other conditioning terms in (25). Removing these other terms, (25) can be rewritten as

$$\Pr\{\alpha_e = a_e | R_e = \tau_e, F_{e+1} < R_e\}. \tag{49}$$

Substituting for α_e from (11) into (49), we get

$$\begin{aligned}
&\Pr\{\alpha_e = a_e | R_e = \tau_e, F_{e+1} < R_e\} \\
&\qquad = \Pr\left\{ \frac{R_e - F_{e+1}}{R_e} = a_e \middle| R_e = \tau_e, F_{e+1} < R_e \right\} && (50)\\
&\qquad = \frac{\Pr\left\{ \frac{R_e - F_{e+1}}{R_e} = a_e, R_e = \tau_e, F_{e+1} < R_e \right\}}{\Pr\{R_e = \tau_e, F_{e+1} < R_e\}} && (51)\\
&\qquad = \frac{\Pr\{F_{e+1} = \tau_e(1-a_e), R_e = \tau_e, F_{e+1} < \tau_e\}}{\Pr\{R_e = \tau_e, F_{e+1} < \tau_e\}} && (52)\\
&\qquad = \frac{\Pr\{F_{e+1} = \tau_e(1-a_e), F_{e+1} < \tau_e\} \Pr\{R_e = \tau_e\}}{\Pr\{F_{e+1} < \tau_e\} \Pr\{R_e = \tau_e\}} && (53)\\
&\qquad = \frac{\Pr\{F_{e+1} = \tau_e(1-a_e)\}}{\Pr\{F_{e+1} < \tau_e\}}. && (54)
\end{aligned}$$

Here, (53) follows from the fact that the time to next node failure, F_{e+1}, and the time to rebuild the most-exposed data, R_e, are independent. From (48), we have

$$\Pr\{F_{e+1} < \tau_e\} \approx \tilde{n}(e)\lambda\tau_{e-1}, \tag{55}$$

and

$$\begin{aligned}
&\Pr\{F_{e+1} = \tau_e(1-a_e)\} \\
&\qquad = \Pr\{\tau_e(1-(a_e+\delta a_e) < F_{e+1} \le \tau_e(1-a_e))\} \\
&\qquad = \Pr\{F_{e+1} \le \tau_e(1-a_e)\} - \Pr\{F_{e+1} \le \tau_e(1-(a_e+\delta a_e))\} \\
&\qquad \approx \tilde{n}(e)\lambda\tau_{e-1}(1-a_e) - \tilde{n}(e)\lambda\tau_{e-1}(1-(a_e+\delta a_e)) && (56)\\
&\qquad = \tilde{n}(e)\lambda\tau_{e-1}\delta a_e, && (57)
\end{aligned}$$

where δa_e denotes an infinitesimal increment of a_e. From (49), (54), (55), and (57), we observe that type B terms of the form (25) can be reduced to (28). □

F Proof of Lemma 3

Type C expressions of the form (26) denote the conditional probability that the rebuild time, R_e, in exposure level e is equal to τ_e. Given that $R_{e-1} = \tau_{e-1}$ and $\alpha_{e-1} = a_{e-1}$, the rebuild time in exposure level e is independent of the other conditioning terms. Removing these other terms, (25) can be rewritten as

$$\Pr\{R_e = \tau_e | R_{e-1} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}\}. \tag{58}$$

Now, given that the rebuild times R_e satisfy (13), the above expression reduces to (29). □