Reliability Assessment of Erasure Coded Systems

Ilias Iliadis and Vinodh Venkatesan
IBM Research – Zurich
8803 Rüschlikon, Switzerland
Email: {ili,ven}@zurich.ibm.com

Abstract—**Replication is widely used to enhance the reliability of storage systems and protect data from device failures. The effectiveness of the replication scheme has been evaluated based on the Mean Time to Data Loss** (MTTDL) **and the Expected Annual Fraction of Data Loss** (EAFDL) **metrics. To provide high data reliability at high storage efficiency, modern systems employ advanced erasure coding redundancy and recovering schemes. This article presents a general methodology for obtaining the** EAFDL **and** MTTDL **of erasure coded systems analytically for the symmetric, clustered and declustered data placement schemes. Our analysis establishes that the declustered placement scheme offers superior reliability in terms of both metrics. The analytical results obtained enable the derivation of the optimal codeword lengths that maximize the** MTTDL **and minimize the** EAFDL.

Keywords–Reliability metric; MTTDL; EAFDL; RAID; MDS codes; Information Dispersal Algorithm; Prioritized rebuild.

I. INTRODUCTION

The reliability of storage systems is affected by data losses due to device and component failures, including disk and node failures. Permanent loss of data is prevented by deploying redundancy schemes that enable data recovery. However, additional device failures that may occur during rebuild operations could lead to permanent data losses. Over the years, several redundancy and recovery schemes have been developed to enhance the reliability of storage systems. These schemes offer different levels of reliability, with varying corresponding overheads due to the additional operations that need to be performed, and different levels of storage efficiencies that depend on the additional amount of redundant (parity) data that needs to be stored in the system.

The effectiveness of the redundancy schemes has been evaluated predominately based on the Mean Time to Data Loss (MTTDL) metric. Closed-form reliability expressions are typically obtained using Markov models with the underlying assumption that the times to component failures and the rebuild times are independent and exponentially distributed [1-13]. Recent work has shown that these results also hold in the practical case of non-exponential failure time distributions. This was achieved based on a methodology for obtaining MTTDL that does not involve any Markov analysis [14]. The MTTDL metric has been used extensively to assess tradeoffs, to compare schemes and to estimate the effect of various parameters on system reliability [15-18].

To cope with data losses encountered in the case of distributed and cloud storage systems, data is replicated and recovery mechanisms are used. For instance, Amazon S3 is designed to provide 99.999999999% (eleven nines) durability of data over a given year [19]. Similarly, Facebook [20], LinkedIn [21] and Yahoo! [22] consider the amount of data lost in given periods. To address this issue, a recent work has introduced the Expected Annual Fraction of Data Loss (EAFDL) metric [23]. It has also presented a methodology for deriving this metric analytically in the case of replication-based storage systems, where user data is replicated r times and the copies are stored in different devices. As an alternative to replication, storage systems use advanced erasure codes that provide a high data reliability as well as a high storage efficiency. The use of such erasure codes can be traced back to as early as the 1980s when they were applied in systems with redundant arrays of inexpensive disks (RAID) [1][2]. The RAID-5, RAID-6 and replication-based systems are special cases of erasure coded systems. State-of-the-art data storage systems [24][25] employ more general erasure codes, where the choice of the codes used greatly affects the performance, reliability, and storage and reconstruction overhead of the system. In this article, we focus on the reliability assessment of erasure coded systems and how the choice of codes affects the reliability in terms of the MTTDL and EAFDL metrics.

The MTTDL of erasure coded systems has been obtained analytically in [26]. To reduce the amount of data lost, it is imperative to assess not only the frequency of data loss events, which is obtained through the MTTDL metric, but also the amount of data lost, which is expressed by the EAFDL metric [23]. The EAFDL and MTTDL metrics provide a useful profile of the size and frequency of data losses. Towards that goal, we present a general framework and methodology for deriving the EAFDL analytically, along with the MTTDL, for the case of erasure coded storage systems. The model developed captures the effect of the various system parameters as well as the effect of various codeword placement schemes, such as clustered, declustered, and symmetric data placement schemes. The results obtained show that the declustered placement scheme offers superior reliability in terms of both metrics. We also investigate the effect of the codeword length and identify the optimal values that offer the best reliability.

The remainder of the paper is organized as follows. Section II describes the storage system model and the corresponding parameters considered. Section III presents the general framework and methodology for deriving the MTTDL and EAFDL metrics analytically for the case of erasure coded systems. Closed-form expressions for the symmetric, clustered, and declustered placement schemes are derived. Section IV compares these schemes and establishes that the declustered placement scheme offers superior reliability. Section V presents a thorough comparison of the reliability achieved by the declustered placement scheme under various codeword configurations. Finally, we conclude in Section VI.

II. STORAGE SYSTEM MODEL

The storage system considered comprises n storage devices (nodes or disks), with each device storing an amount c of data,

 ISBN: 978-1-61208-550-0

TABLE I. NOTATION OF SYSTEM PARAMETERS

Parameter	Definition
n	number of storage devices
c	amount of data stored on each device
l	number of user-data symbols per codeword ($l \geq 1$)
m	total number of symbols per codeword ($m > l$)
(l, m)	MDS-code structure
s	symbol size
k	spread factor of the data placement scheme
b	reserved rebuild bandwidth per device
$1/\lambda$	mean time to failure of a storage device
s_{eff}	storage efficiency of redundancy scheme ($s_{\text{eff}} = l/m$)
U	amount of user data stored in the system ($U = s_{\text{eff}}\, n\, c$)
$\tilde{r}$	minimum number of codeword symbols lost that lead to an irrecoverable data loss ($\tilde{r} = m - l + 1$ and $2 \leq \tilde{r} \leq m$)
$1/\mu$	time to read (or write) an amount c of data at a rate b from (or to) a device ($1/\mu = c/b$)

such that the total storage capacity of the system is $n\, c$. Modern data storage systems use various forms of data redundancy to protect data from device failures. When devices fail, the redundancy of the data affected is reduced and eventually lost. To avoid irrecoverable data loss, the system performs rebuild operations that use the data stored in the surviving devices to reconstruct the temporarily lost data, thus maintaining the initial data redundancy.

A. Redundancy

According to the erasure coded schemes considered, the user data is divided into blocks (or symbols) of a fixed size (e.g., sector size of 512 bytes) and complemented with parity symbols to form codewords. In this article, we consider (l, m) maximum distance separable (MDS) erasure codes, which are a mapping from l user data symbols to a set of m ($> l$) symbols, called a codeword, in such a way that any subset containing l of the m symbols of the codeword can be used to decode (reconstruct, recover) the codeword. The corresponding storage efficiency, s_{eff}, is given by

$$s_{\text{eff}} = \frac{l}{m} \, , \tag{1}$$

such that the amount of user data, U, stored in the system is given by

$$U = s_{\text{eff}}\, n\, c \; = \; \frac{l\, n\, c}{m} \, . \tag{2}$$

The notation used is summarized in Table I. The parameters are divided according to whether they are independent or derived, and are listed in the upper and the lower part of the table, respectively.

The m symbols of each codeword are stored on m distinct devices, such that the system can tolerate any $\tilde{r} - 1$ device failures, but a number of $\tilde{r}$ device failures may lead to data loss, with

$$\tilde{r} = m - l + 1 \, . \tag{3}$$

From the preceding, it follows that

$$1 \leq l < m \qquad \text{and} \qquad 2 \leq \tilde{r} \leq m \, . \tag{4}$$

Examples of MDS erasure codes are the following:

Replication: A replication-based system with a replication factor r can tolerate any loss of up to $r - 1$ copies of some data, such that $l = 1$, $m = r$ and $\tilde{r} = r$. Also, its storage efficiency is equal to $s_{\text{eff}}^{(\text{replication})} = 1/r$.
RAID-5: A RAID-5 array comprised of N devices uses an $(N-1, N)$-MDS code, such that $l = N - 1$, $m = N$ and $\tilde{r} = 2$. It can therefore tolerate the loss of up to one device, and its storage efficiency is equal to $s_{\text{eff}}^{(\text{RAID-5})} = (N-1)/N$.
RAID-6: A RAID-6 array comprised of N devices uses an $(N-2, N)$-MDS code, such that $l = N - 2$, $m = N$ and $\tilde{r} = 3$. It can therefore tolerate a loss of up to two devices, and its storage efficiency is equal to $s_{\text{eff}}^{(\text{RAID-6})} = (N-2)/N$.
Reed–Solomon: It is based on (l, m)-MDS erasure codes.

B. Symmetric Codeword Placement

We consider a placement where each codeword is stored on m distinct devices with one symbol per device. In a large storage system, the number of devices, n, is typically much larger than the codeword length, m. Therefore, there exist many ways in which a codeword of m symbols can be stored across a subset of the n devices. For each device in the system, let its *redundancy spread factor* k denote the number of devices over which the codewords stored on that device are spread [26]. In a symmetric placement scheme, the $m-1$ symbols of each codeword corresponding to the data on each device are *equally* spread across $k - 1$ other devices, such that these devices alltogether form a group of k devices. It also holds that the $m-2$ codeword symbols corresponding to the codewords shared by any two devices within this group are equally spread across $k - 2$ other devices, and so on. Consequently, all the symbols of each codeword in the system are contained within such a group, which implies that the system is effectively comprised of n/k disjoint groups of k devices. Each group contains an amount U/k of user data, with the corresponding codewords placed on the corresponding k devices in a distributed manner. We proceed by considering the clustered and declustered placement schemes, which are special cases of symmetric placement schemes for which k is equal to m and n, respectively.

1) Clustered Placement: In this placement scheme, the n devices are divided into disjoint sets of m devices, referred to as *clusters*. According to the *clustered* placement, each codeword is stored across the devices of a particular cluster. In such a placement scheme, it can be seen that no cluster stores the redundancies that correspond to data stored on another cluster. The entire storage system can essentially be modeled as consisting of n/m independent clusters. In each cluster, data loss occurs when $\tilde{r}$ devices fail successively before rebuild operations complete successfully.

2) Declustered Placement: In this placement scheme, all $\binom{n}{m}$ possible ways of placing m symbols across n devices are equally used to store all the codewords in the system. This is a symmetric placement scheme, in that for any given device, the same number of the codeword symbols that correspond to the codewords on that device are contained in each set of any other $m-1$ devices.

C. Codeword Reconstruction

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

1) Exposure Levels and Amount of Data to Rebuild: At time t, let $D_j(t)$ be the number of codewords that have lost j symbols, with $0 \leq j \leq \tilde{r}$. The system is at exposure level e $(0 \leq e \leq \tilde{r})$, where

$$e = \max_{D_j(t)>0} j. \tag{5}$$

In other words, the system is at exposure level e if there are codewords with $m-e$ symbols left, but there are no codewords with fewer than $m - e$ symbols left in the system, that is, $D_e(t) > 0$, and $D_j(t) = 0$ for all $j > e$. These codewords are referred to as the *most-exposed* codewords. Let the number of most-exposed codewords when entering exposure level e be denoted by C_e, $e = 1, \ldots, \tilde{r}$. At $t = 0$, $D_j(0) = 0$ for all $j > 0$ and $D_0(0)$ is the total number of codewords stored in the system. Device failures and rebuild processes cause the values of $D_1(t), \cdots, D_{\tilde{r}}(t)$ to change over time, and when a data loss occurs, $D_{\tilde{r}}(t) > 0$. Device failures cause transitions to higher exposure levels, whereas rebuilds cause transitions to lower ones.

In this article, we will derive the reliability metrics of interest using the direct path approximation, which considers only transitions from lower to higher exposure levels [14][26][27]. This implies that each exposure level is entered only once.

2) Prioritized or Intelligent Rebuild: At each exposure level e, the *prioritized or intelligent* rebuild process attempts to bring the system back to exposure level $e-1$ by recovering one of the e symbols that each of the most-exposed codewords has lost, that is, by recovering a total number of C_e symbols. Let A_e denote the amount of data corresponding to the C_e symbols and let s denote the symbol size. Then, it holds that

$$A_e = C_e \, s \, . \tag{6}$$

The notation used is summarized in Table II. For an exposure level e $(< \tilde{r})$, A_e represents the amount of data that needs to be rebuilt at that exposure level. In particular, upon the first-device failure, it holds that

$$A_1 = c \, . \tag{7}$$

D. Rebuild Process

During the rebuild process, a certain proportion of the device bandwidth is reserved for data recovery, with b denoting the actual reserved rebuild bandwidth per device. The rebuild bandwidth is usually only a fraction of the total bandwidth available at each device; the remainder is used to serve user requests. Let us denote by b_e $(\leq b)$ the rate at which the amount A_e of data that needs to be rebuilt at exposure level e is written to selected device(s). In particular, let us denote by $1/\mu$ the time required to read (or write) an amount c of data from (or to) a device, given by

$$\frac{1}{\mu} = \frac{c}{b} \, . \tag{8}$$

E. Failure and Rebuild Time Distributions

In this work, we assume that the lifetimes of the n devices are independent and identically distributed, with a cumulative distribution function $F_\lambda(.)$ and a mean of $1/\lambda$. We further consider storage devices with failure time distributions that belong to the large class defined in [14], which includes real-world distributions, such as Weibull and gamma, as well as exponential distributions. The storage devices are *highly reliable* when the ratio of the fixed time $1/\mu$ to read all contents of a device (which typically is on the order of tens of hours) to the mean time to failure of a device $1/\lambda$ (which is typically at least on the order of thousands of hours) is small, that is, when

$$\frac{\lambda}{\mu} = \frac{\lambda \, c}{b} \ll 1 \, . \tag{9}$$

According to [14][26], when the cumulative distribution function F_λ satisfies the condition

$$\mu \int_0^{1/\mu} F_\lambda(t) \, dt \ll 1, \quad \text{with } \frac{\lambda}{\mu} \ll 1 \, , \tag{10}$$

the MTTDL reliability metric of replication-based or erasure coded storage systems tends to be insensitive to the device failure distribution, that is, the MTTDL depends only on its mean $1/\lambda$, but not on its density $F_\lambda(.)$. In [23], it was shown that this also holds for the EAFDL metric in the case of replication-based storage systems, and in this article, we will show that this is also the case for erasure coded systems.

TABLE II. NOTATION OF SYSTEM PARAMETERS AT EXPOSURE LEVELS

Parameter	Definition
e	exposure level
C_e	number of most-exposed codewords when entering exposure level e
R_e	rebuild time at exposure level e
$P_{e \to e+1}$	transition probability from exposure level e to $e+1$
$\tilde{n}_e$	number of devices at exposure level e whose failure causes an exposure level transition to level $e+1$
α_e	fraction of the rebuild time R_e still left when another device fails causing the exposure level transition $e \to e+1$
V_e	fraction of the most-exposed codewords that have symbols stored on another of the $\tilde{n}_e$ devices
A_e	amount of data corresponding to the C_e symbols $(A_e = C_e \, s)$
b_e	rate at which recovered data is written at exposure level e

III. DERIVATION OF MTTDL AND EAFDL

We briefly review the general methodology for deriving the MTTDL and EAFDL metrics presented in [23]. This methodology does not involve any Markov analysis and holds for general failure time distributions, which can be exponential or non-exponential, such as the Weibull and gamma distributions.

At any point of time, the system can be thought to be in one of two modes: normal mode and rebuild mode. During normal mode, all data in the system has the original amount of redundancy and there is no active rebuild process. During 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 normal mode to rebuild mode occurs when a device fails; we refer to the device failure that causes this transition as a *first-device* failure. Following a first-device failure, a complex sequence of rebuild operations and subsequent device failures may occur, which eventually leads the system either to an irrecoverable data loss (DL) with probability P_{DL} or back to the original normal mode by restoring initial redundancy, which occurs with probability $1 - P_{\text{DL}}$. The MTTDL is then given by [23]

$$\text{MTTDL} \approx \frac{1}{n \, \lambda \, P_{\text{DL}}} \, . \tag{11}$$

 ISBN: 978-1-61208-550-0

Let H denote the corresponding amount of data lost conditioned on the fact that a data loss has occurred. The metric of interest, that is, the Expected Annual Fraction of Data Loss (EAFDL), is subsequently obtained as the ratio of the expected amount of data lost to the expected time to data loss normalized to the amount of user data:

$$\text{EAFDL} = \frac{E(H)}{\text{MTTDL} \cdot U} , \tag{12}$$

with the MTTDL expressed in years. Let us also denote by Q the unconditional amount of data lost upon a first-device failure. Note that Q is unconditional on the event of a data loss occurring in that it is equal either to H if the system suffers a data loss prior to returning to normal operation or to zero otherwise, that is,

$$Q = \begin{cases} H , & \text{if DL} \\ 0 , & \text{if no DL} . \end{cases} \tag{13}$$

Therefore, the expected amount of data lost, $E(Q)$, upon a first-device failure is given by

$$E(Q) = P_{\text{DL}} \, E(H) . \tag{14}$$

From (11), (12) and (14), we obtain the EAFDL as follows:

$$\text{EAFDL} \approx \frac{n \, \lambda \, E(Q)}{U} , \tag{15}$$

with $1/\lambda$ expressed in years.

A. Reliability Analysis

From (11) and (15), it follows that the derivation of the MTTDL and EAFDL metrics requires the evaluation of the P_{DL} and $E(Q)$, respectively. These quantities are derived by considering the direct path approximation [12][13][27], which, under conditions (9) and (10), accurately assesses the reliability metrics of interest [14][23].

Next, we present the general outline of the methodology in more detail.

1) Direct Path to Data Loss: Consider the direct path of successive transitions from exposure level 1 to $\tilde{r}$. In [12][13][27], it was shown that P_{DL} can be approximated by the probability of the direct path to data loss, $P_{\text{DL,direct}}$, that is,

$$P_{\text{DL}} \approx P_{\text{DL,direct}} = \prod_{e=1}^{r-1} P_{e \to e+1}, \tag{16}$$

where $P_{e \to e+1}$ denotes the transition probability from exposure level e to $e+1$. The above approximation holds when storage devices are highly reliable, that is, it holds for arbitrary device failure and rebuild time distributions that satisfy conditions (9) and (10). In this case, the relative error tends to zero as λ/μ tends to zero [14].

As the direct path to data loss dominates the effect of all other possible paths to data loss considered together, it follows that the amount of data lost H can be approximated by that corresponding to the direct path:

$$H \approx H_{\text{direct}} . \tag{17}$$

Also, from (13) and (17) it follows that

$$Q \approx \begin{cases} H_{\text{direct}} , & \text{if DL follows the direct path} \\ 0 , & \text{otherwise} . \end{cases} \tag{18}$$

Consequently, to derive the amount of data lost, it suffices to proceed by considering the H and Q metrics corresponding to the direct path to data loss.

Note that the amount of data lost, H, is the amount of user data stored in the most-exposed codewords when entering exposure level $\tilde{r}$, which can no longer be recovered and therefore is irrecoverably lost. As the number of these codewords is equal to $C_{\tilde{r}}$ and each of these codewords contains l symbols of user data, it holds that

$$H = C_{\tilde{r}} \, l \, s \overset{(6)}{=} l \, A_{\tilde{r}} . \tag{19}$$

2) Amount of Data to Rebuild and Rebuild Times at Each Exposure Level: We now proceed to derive the conditional values of the random variables of interest given that the system goes through this direct path to data loss. Let R_e denote the rebuild times of the most-exposed codewords at each exposure level in this path, and let α_e be the fraction of the rebuild time R_e still left when another device fails causing the exposure level transition $e \to e+1$. In [28, Lemma 2], it was shown that, for highly reliable devices satisfying conditions (9) and (10), α_e is approximately uniformly distributed between zero and one, that is,

$$\alpha_e \sim U(0,1), \quad e = 1, \ldots, \tilde{r}-1 . \tag{20}$$

Let $\vec{\alpha}$ denote the vector $(\alpha_1, \ldots, \alpha_{\tilde{r}-1})$, $\vec{\alpha_e}$ the vector $(\alpha_1, \ldots, \alpha_e)$, $\vec{C_e}$ the vector $(C_1, \cdots, C_e)$ and $\vec{A_e}$ the vector $(A_1, \cdots, A_e)$. Clearly, for the rebuild schemes considered, the fraction α_e of the rebuild time R_e still left also represents the fraction of the most-exposed codewords not yet recovered upon the next device failure. Therefore, the number of most-exposed codewords that are not yet recovered is equal to $\alpha_e \, C_e$. Clearly, the fraction V_e of these codewords that have symbols stored on the newly failed device depends on the codeword placement scheme. Consequently, the number of most-exposed codewords when entering exposure level $e+1$ is given by

$$C_{e+1} = V_e \, \alpha_e \, C_e, \, , \quad e = 1, \ldots, \tilde{r}-1 , \tag{21}$$

and by virtue of (6), the corresponding amount of data that is not yet rebuilt is given by

$$A_{e+1} = V_e \, \alpha_e \, A_e, \, , \quad e = 1, \ldots, \tilde{r}-1 , \tag{22}$$

with V_e depending only on the placement scheme.

Repeatedly applying (22) and using (7) yields

$$A_e = c \prod_{j=1}^{e-1} V_j \, \alpha_j . \tag{23}$$

Remark 1: From (23), it follows that the expected amount of data to be rebuilt at each exposure level does not depend on the duration of the rebuild times.

At exposure level 1, according to (7), the amount A_1 of data to be recovered is equal to c. Given that this data is recovered at a rate of b_1 and that the time required to write an amount c of data at a rate of b is equal to $1/\mu$, it follows that the rebuild time R_1 is given by

$$R_1 = \frac{b}{b_1} \cdot \frac{1}{\mu} . \tag{24}$$

 ISBN: 978-1-61208-550-0

As the rebuild times are proportional to the amount of data to be rebuilt and are inversely proportional to the rebuild rates, it holds that

$$\frac{R_{e+1}}{R_e} = \frac{A_{e+1}}{A_e} \cdot \frac{b_e}{b_{e+1}} , \quad e \geq 1 . \tag{25}$$

Using (22), (25) yields

$$R_{e+1} = V_e \, \alpha_e \, \frac{b_e}{b_{e+1}} \, R_e \, , \quad e = 1, \ldots, \tilde{r} - 2 \, , \tag{26}$$

or

$$R_e = G_{e-1} \, \alpha_{e-1} \, R_{e-1} \, , \quad e = 2, \ldots, \tilde{r} - 1 \, , \tag{27}$$

where

$$G_e \triangleq \frac{b_e}{b_{e+1}} \, V_e \, , \quad e = 1, \ldots, \tilde{r} - 2 \, . \tag{28}$$

Repeatedly applying (27) and using (28) yields

$$R_e = \frac{b_1}{b_e} \, R_1 \prod_{j=1}^{e-1} V_j \, \alpha_j \, , \quad e = 1, \ldots, \tilde{r} - 1 \, . \tag{29}$$

Let $\tilde{n}_e$ be the number of devices at exposure level e whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $e+1$. Subsequently, the transition probability $P_{e \to e+1}$ from exposure level e to $e+1$ depends on the duration of the corresponding rebuild time R_e and the aggregate failure rate of these $\tilde{n}_e$ highly reliable devices, and is given by [14]

$$P_{e \to e+1} \approx \tilde{n}_e \, \lambda \, R_e \, , \quad \text{for } e = 1, \ldots, \tilde{r} - 1 \, . \tag{30}$$

Substituting (29) into (30) yields

$$P_{e \to e+1}(\vec{\alpha}_{e-1}) \approx \tilde{n}_e \, \lambda \, \frac{b_1}{b_e} \, R_1 \prod_{j=1}^{e-1} V_j \, \alpha_j \, . \tag{31}$$

3) Estimation of P_{DL}: Consider the direct path $1 \to 2 \to \cdots \to \tilde{r}$ of successive transitions from exposure level 1 to $\tilde{r}$. For ease of reading, we denote the successive transitions from exposure level e to $\tilde{r}$ by $e \to \tilde{r}$. We first evaluate P_{DL}, the probability of data loss. From (16), and using the fact that α_e does not depend on $R_1, \alpha_1, \cdots, \alpha_{e-1}$, it follows that

$$\begin{aligned}
P_{\mathrm{DL}} \; &\approx \; P_{1 \to \tilde{r}} = P_{1 \to 2} P_{2 \to \tilde{r}} = P_{1 \to 2} E_{\alpha_1}[P_{2 \to \tilde{r}}(\alpha_1)] \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\alpha_1) P_{3 \to \tilde{r}}(\alpha_1)] \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\alpha_1) E_{\alpha_2 | \alpha_1}[P_{3 \to \tilde{r}}(\alpha_1, \alpha_2)]] \\
&= \; \cdots \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\vec{\alpha}_1) E_{\alpha_2}[P_{3 \to 4}(\vec{\alpha}_2) \; \cdots \\
&\qquad \cdots \; E_{\alpha_{\tilde{r}-2}}[P_{\tilde{r}-1 \to \tilde{r}}(\vec{\alpha}_{\tilde{r}-2})] \cdots] \\
&= E_{\vec{\alpha}_{\tilde{r}-2}}[P_{1 \to 2} P_{2 \to 3}(\vec{\alpha}_1) \; \cdots \; P_{\tilde{r}-1 \to \tilde{r}}(\vec{\alpha}_{\tilde{r}-2})] \\
&= E_{\vec{\alpha}_{\tilde{r}-2}}\left[\prod_{e=1}^{\tilde{r}-1} P_{e \to e+1}(\vec{\alpha}_{e-1})\right] = E_{\vec{\alpha}_{\tilde{r}-2}}[P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2})] \, ,
\end{aligned} \tag{32}$$

where

$$P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \; \triangleq \; \prod_{e=1}^{\tilde{r}-1} P_{e \to e+1}(\vec{\alpha}_{e-1}) \, , \tag{33}$$

with

$$P_{1 \to 2}(\vec{\alpha}_0) \triangleq P_{1 \to 2} \, . \tag{34}$$

Substituting (31) into (33), and using (30) and (34), yields

$$P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \approx (\lambda \, b_1 \, R_1)^{\tilde{r}-1} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, (V_e \, \alpha_e)^{\tilde{r}-1-e} \, . \tag{35}$$

Unconditioning (35) on $\vec{\alpha}_{\tilde{r}-2}$, and given that the elements of $\vec{\alpha}_{\tilde{r}-2}$ are independent random variables approximately distributed according to (20) such that $E(\alpha_e^k) \approx 1/(k+1)$, (32) yields

$$P_{\mathrm{DL}} \approx (\lambda \, b_1 \, R_1)^{\tilde{r}-1} \, \frac{1}{(\tilde{r}-1)!} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, V_e^{\tilde{r}-1-e} \, . \tag{36}$$

Using (8) and (24), (36) yields

$$P_{\mathrm{DL}} \approx (\lambda \, c)^{\tilde{r}-1} \, \frac{1}{(\tilde{r}-1)!} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, V_e^{\tilde{r}-1-e} \, . \tag{37}$$

4) Estimation of $E(Q)$: We now proceed to evaluate $E(Q)$, the expected amount of data lost. Considering the *direct path* $1 \to 2 \to \cdots \to \tilde{r}$ of successive transitions from exposure level 1 to $\tilde{r}$, it follows from (18) and the fact that α_e does not depend on $R_1, \alpha_1, \cdots, \alpha_{e-1}$, that

$$\begin{aligned}
E(Q) \; &\approx \; P_{1 \to 2} E(Q | 1 \to 2) \\
&= P_{1 \to 2} E_{\alpha_1 | R_1}[E(Q | \alpha_1)] \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\alpha_1) E(Q | \alpha_1, 2 \to 3)] \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\alpha_1) E_{\alpha_2 | \alpha_1}[E(Q | \alpha_1, \alpha_2)]] \\
&= \; \cdots \\
&= P_{1 \to 2} E_{\alpha_1}[P_{2 \to 3}(\vec{\alpha}_1) E_{\alpha_2}[P_{3 \to 4}(\vec{\alpha}_2) \; \cdots \\
&\qquad \cdots \; P_{\tilde{r}-1 \to \tilde{r}}(\vec{\alpha}_{\tilde{r}-2}) E_{\alpha_{\tilde{r}-1}}(Q | \vec{\alpha}_{\tilde{r}-1})] \cdots] \\
&= E_{\vec{\alpha}_{\tilde{r}-1}}[P_{1 \to 2} P_{2 \to 3}(\vec{\alpha}_1) \; \cdots \; P_{\tilde{r}-1 \to \tilde{r}}(\vec{\alpha}_{\tilde{r}-2}) \\
&\qquad\qquad E(Q | \vec{\alpha}_{\tilde{r}-1})] \\
&\overset{(17)(18)}{=} E_{\vec{\alpha}_{\tilde{r}-1}}\left[\left(\prod_{e=1}^{\tilde{r}-1} P_{e \to e+1}(\vec{\alpha}_{e-1})\right) E(H | \vec{\alpha}_{\tilde{r}-1})\right] \\
&\overset{(33)}{=} E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \, E(H | \vec{\alpha}_{\tilde{r}-1})] \\
&\overset{(19)}{=} E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \, E(l \, A_{\tilde{r}} | \vec{\alpha}_{\tilde{r}-1})] \\
&= E_{\vec{\alpha}_{\tilde{r}-1}}[P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \, l \, E(A_{\tilde{r}} | \vec{\alpha}_{\tilde{r}-1})] \\
&= E_{\vec{\alpha}_{\tilde{r}-1}}[G(\vec{\alpha}_{\tilde{r}-1})] \, ,
\end{aligned} \tag{38}$$

where

$$G(\vec{\alpha}_{\tilde{r}-1}) \; \triangleq \; l \, P_{\mathrm{DL}}(\vec{\alpha}_{\tilde{r}-2}) \, E(A_{\tilde{r}} | \vec{\alpha}_{\tilde{r}-1}) \, . \tag{39}$$

Using (23) and (35), (39) yields

$$G(\vec{\alpha}_{\tilde{r}-1}) \; \approx \; l \, c \, (\lambda \, b_1 \, R_1)^{\tilde{r}-1} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, (V_e \, \alpha_e)^{\tilde{r}-e} \, . \tag{40}$$

Unconditioning (40) on $\vec{\alpha}_{\tilde{r}-1}$, and given that the elements of $\vec{\alpha}_{\tilde{r}-1}$ are independent random variables approximately distributed according to (20) such that $E(\alpha_e^k) \approx 1/(k+1)$, (38) yields

$$E(Q) \approx l \, c \, (\lambda \, b_1 \, R_1)^{\tilde{r}-1} \, \frac{1}{\tilde{r}\,!} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, V_e^{\tilde{r}-e} \, . \tag{41}$$

Using (8) and (24), (41) yields

$$E(Q) \approx l \, c \, (\lambda \, c)^{\tilde{r}-1} \, \frac{1}{\tilde{r}\,!} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, V_e^{\tilde{r}-e} \, . \tag{42}$$

5) Evaluation of $E(H)$: The expected amount $E(H)$ of data lost conditioned on the fact that a data loss has occurred is obtained from (14) as the ratio of $E(Q)$ to $P_{\rm DL}$. Consequently, using (37) and (42), it follows that

$$E(H) = \frac{E(Q)}{P_{\rm DL}} \approx \left(\frac{l}{\tilde{r}} \prod_{e=1}^{\tilde{r}-1} V_e \right) c \ . \tag{43}$$

Remark 2: From (43), it follows that the expected amount of data lost conditioned on the fact that a data loss has occurred does not depend on the duration of the rebuild times.

6) Evaluation of MTTDL *and* EAFDL*:* Substituting (37) into (11) yields

$$\text{MTTDL} \approx \frac{1}{n\,\lambda} \, \frac{(\tilde{r}-1)!}{(\lambda\, c)^{\tilde{r}-1}} \prod_{e=1}^{\tilde{r}-1} \frac{b_e}{\tilde{n}_e} \, \frac{1}{V_e^{\tilde{r}-1-e}} \ . \tag{44}$$

Substituting (2) and (42) into (15) yields

$$\text{EAFDL} \approx m\,\lambda\,(\lambda\,c)^{\tilde{r}-1} \, \frac{1}{\tilde{r}\,!} \prod_{e=1}^{\tilde{r}-1} \frac{\tilde{n}_e}{b_e} \, V_e^{\tilde{r}-e} \ . \tag{45}$$

B. Symmetric Scheme

Here, we consider the case where the redundancy spread factor k is in the interval $m < k \le n$. As discussed in Section II-C2, at each exposure level e, the *prioritized* rebuild process recovers one of the e symbols that each of the C_e most-exposed codewords has lost by reading $m - \tilde{r} + 1$ of the remaining symbols. Thus, there are C_e symbols to be recovered in total, which corresponds to an amount A_e of data. For the symmetric placement discussed in Section II-B, these symbols are recovered by reading $(m - \tilde{r} + 1)\, C_e$ symbols, which corresponds to an amount $(m - \tilde{r} + 1)\, A_e$ of data, from the $k - e$ surviving devices in the affected group. Note that these are precisely the devices at exposure level e whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $e + 1$. Consequently, it holds that

$$\tilde{n}_e^{\rm sym} = k - e \ . \tag{46}$$

Furthermore, the recovered symbols are written to the spare space of these devices in such a way that no symbol is written to a device in which another symbol corresponding to the same codeword is already present. Owing to the symmetry of the symmetric placement, the same amount of data is being read from each of the $\tilde{n}_e$ devices. Similarly, the same amount of data is being written to each of the $\tilde{n}_e$ devices. Consequently, the total read/write rebuild bandwidth b of each device is split between the reads and the writes, with the read rate being equal to $(m - \tilde{r} + 1)\, b/(m - \tilde{r} + 2)$ and the write rate being equal to $b/(m - \tilde{r} + 2)$. Therefore, the total write bandwidth, which is also the rebuild rate b_e, is given by

$$b_e^{\rm sym} = \frac{\tilde{n}_e^{\rm sym}}{m - \tilde{r} + 2} \, b \, , \quad e = 1, \ldots, \tilde{r} - 1 \ . \tag{47}$$

Once all lost codeword symbols have been recovered, they are transferred to a new replacement device.

When the system enters exposure level e, the number of most-exposed codewords that need to be recovered is equal to C_e, $e = 1, \ldots, \tilde{r}$. Upon the next device failure, the expected number of most-exposed codewords that are not yet recovered is equal to $\alpha_e\, C_e$. Owing to the nature of the symmetric codeword placement, the newly failed device stores codeword symbols corresponding to only a fraction

$$V_e^{\rm sym} = \frac{m - e}{k - e} \ , \quad e = 1, \ldots, \tilde{r} - 1 \ . \tag{48}$$

of these most-exposed, not yet recovered codewords.

Substituting (46), (47) and (48) into (44), (45) and (43), and using (3) yields

$$\text{MTTDL}_k^{\rm sym} \approx \frac{1}{n\,\lambda} \left[\frac{b}{(l+1)\,\lambda\,c} \right]^{m-l} (m-l)! \prod_{e=1}^{m-l} \left(\frac{k-e}{m-e} \right)^{m-l-e} \ , \tag{49}$$

$$\text{EAFDL}_k^{\rm sym} \approx \lambda \left[\frac{(l+1)\,\lambda\,c}{b} \right]^{m-l} \frac{m}{(m-l+1)!} \prod_{e=1}^{m-l} \left(\frac{m-e}{k-e} \right)^{m-l+1-e} \ , \tag{50}$$

and

$$E(H)_k^{\rm sym} \approx \left(\frac{l}{m-l+1} \prod_{e=1}^{m-l} \frac{m-e}{k-e} \right) c \tag{51}$$

$$= \frac{l\,(m-1)!\,(k-m+l-1)!}{(m-l+1)\,(k-1)!\,(l-1)!} \, c \ . \tag{52}$$

Note that for a replication-based system, for which $m = r$ and $l = 1$, (49) and (50) are in agreement with (42.b) and (43.b) of [23], respectively.

Remark 3: From (49), (50), and (51), it follows that $\text{MTTDL}_k^{\rm sym}$ depends on n, but $\text{EAFDL}_k^{\rm sym}$ and $E(H)_k^{\rm sym}$ do not.

Remark 4: From (49), (50), and (51), it follows that, for $m - l = 1$, $\text{MTTDL}_k^{\rm sym}$ does not depend on k, whereas for $m - l > 1$, $\text{MTTDL}_k^{\rm sym}$ is increasing in k. Also, for $m - l \ge 1$, $\text{EAFDL}_k^{\rm sym}$ and $E(H)_k^{\rm sym}$ are decreasing in k. Consequently, within the class of symmetric placement schemes considered, that is, for $m < k \le n$, the $\text{MTTDL}_k^{\rm sym}$ is maximized and the $\text{EAFDL}_k^{\rm sym}$ and the $E(H)_k^{\rm sym}$ are minimized when $k = n$.

C. Clustered Placement

As discussed in Section II-B1, in the clustered placement scheme, the n devices are divided into disjoint sets of m devices, referred to as *clusters*. According to the *clustered* placement, each codeword is stored across the devices of a particular cluster. At each exposure level e, the rebuild process recovers one of the e symbols that each of the C_e most-exposed codewords has lost by reading $m - \tilde{r} + 1$ of the remaining symbols. Note that the remaining symbols are stored on the $m - e$ surviving devices in the affected group. As these are precisely the devices at exposure level e whose failure before the rebuild of the most-exposed codewords causes an exposure level transition to level $e + 1$, it holds that

$$\tilde{n}_e^{\rm clus} = m - e \ . \tag{53}$$

 ISBN: 978-1-61208-550-0

The rebuild process in clustered placement recovers the lost symbols by reading $m-\tilde{r}+1$ symbols from $m-\tilde{r}+1$ of the $\tilde{n}_e$ surviving devices of the affected cluster. The lost symbols are computed on-the-fly and written to a spare device using the rebuild bandwidth at a rate of b. Consequently, it holds that

$$b_e^{\text{clus}} = b\,, \quad e = 1,\ldots,\tilde{r}-1\,. \tag{54}$$

When the system enters exposure level e, the number of most-exposed codewords that need to be recovered is equal to C_e, $e = 1,\ldots,\tilde{r}$. Upon the next device failure, the expected number of most-exposed codewords that have not yet been recovered is equal to $\alpha_e\, C_e$. Clearly, all these codewords have symbols stored on the newly failed device, which implies that

$$V_e^{\text{clus}} = 1\,, \quad e = 1,\ldots,\tilde{r}-1\,. \tag{55}$$

Substituting (53), (54) and (55) into (44), (45) and (43), and using (3) yields

$$\text{MTTDL}^{\text{clus}} \approx \frac{1}{n\,\lambda}\left(\frac{b}{\lambda\,c}\right)^{m-l}\frac{1}{\binom{m-1}{l-1}}\,, \tag{56}$$

$$\text{EAFDL}^{\text{clus}} \approx \lambda\left(\frac{\lambda\,c}{b}\right)^{m-l}\binom{m}{l-1}\,, \tag{57}$$

and

$$E(H)^{\text{clus}} = \frac{l}{m-l+1}\,c\,. \tag{58}$$

Note that for a replication-based system, for which $m = r$ and $l = 1$, (56), (57) and (58) are in agreement with (42.a), (43.a) and (39.a) of [23], respectively.

D. Declustered Placement

As discussed in Section II-B, the declustered placement scheme is a special cases of a symmetric placement scheme in which k is equal to n. Consequently, for $k = n$, (49), (50) and (51) yield

$$\text{MTTDL}^{\text{declus}} \approx \frac{1}{n\,\lambda}\left[\frac{b}{(l+1)\,\lambda\,c}\right]^{m-l}(m-l)!\prod_{e=1}^{m-l}\left(\frac{n-e}{m-e}\right)^{m-l-e}\,, \tag{59}$$

$$\text{EAFDL}^{\text{declus}} \approx \lambda\left[\frac{(l+1)\,\lambda\,c}{b}\right]^{m-l}\frac{m}{(m-l+1)!}\prod_{e=1}^{m-l}\left(\frac{m-e}{n-e}\right)^{m-l+1-e}\,, \tag{60}$$

and

$$E(H)^{\text{declus}} \approx \left(\frac{l}{m-l+1}\prod_{e=1}^{m-l}\frac{m-e}{n-e}\right)c \tag{61}$$

$$= \frac{l\,(m-1)!\,(n-m+l-1)!}{(m-l+1)\,(n-1)!\,(l-1)!}\,c\,. \tag{62}$$

Note that for a replication-based system, for which $m = r$ and $l = 1$, (59), (60) and (61) are in agreement with (36.b), (37.b) and (39.b) of [23], respectively.

IV. Optimal Placement

Here, we identify which of the placement schemes considered offers the best reliability in terms of the MTTDL, EAFDL and $E(H)$ metrics. From Remark 4, it follows that the placement that maximizes MTTDL and minimizes EAFDL and $E(H)$ is either the clustered $(k = m)$ or the declustered one $(k = n)$. We therefore proceed by comparing these two schemes when $m < n$, or, by also using (4), when

$$1 \le l < m \quad \text{and} \quad 1 \le m-l < m < n\,. \tag{63}$$

A. Maximizing MTTDL

From (56) and (59), it follows that

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \left(\frac{1}{l+1}\right)^{m-l}(m-l)!\binom{m-1}{l-1}\prod_{e=1}^{m-l}\left(\frac{n-e}{m-e}\right)^{m-l-e}\,. \tag{64}$$

Remark 5: From (64), it follows that the placement that maximizes MTTDL does not depend on λ, b and c.

Depending on the values of m and l, we consider the following three cases:

1) $m-l=1$: For $m-l=1$, (64) yields

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \frac{m-1}{m} < 1\,. \tag{65}$$

2) $m-l=2$: For $m-l=2$, (64) yields

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \frac{(m-2)(n-1)}{(m-1)^2}\begin{cases}<1 & \text{for } n = m+1\\ >1 & \text{for } n \ge m+2\,.\end{cases} \tag{66}$$

3) $m-l\ge 3$: For $m-l\ge 3$, (64) can be written as follows:

$$\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} \approx \frac{m-1}{l+1}\cdots\frac{l+1}{l+1}\,\frac{l}{l+1}\,\frac{n-m+l+1}{l+1}\left(\frac{n-m+l+2}{l+2}\right)^2\prod_{e=1}^{m-l-3}\left(\frac{n-e}{m-e}\right)^{m-l-e}\,. \tag{67}$$

Using (63), (67) yields

$$\begin{aligned}\frac{\text{MTTDL}^{\text{declus}}}{\text{MTTDL}^{\text{clus}}} &> \frac{l}{l+1}\,\frac{n-m+l+1}{l+1}\left(\frac{n-m+l+2}{l+2}\right)^2\\ &\ge \frac{l}{l+1}\,\frac{l+2}{l+1}\left(\frac{l+3}{l+2}\right)^2 = \frac{l\,(l+3)^2}{(l+1)^2\,(l+2)}\\ &= \frac{2[l^2+2(l-1)+1]}{(l+1)^2\,(l+2)}+1>1\,.\end{aligned} \tag{68}$$

Remark 6: From the preceding, it follows that the MTTDL is maximized by the declustered placement scheme, except in the cases of $m-l=1$ and of $m-l=2$ with $n=m+1$, where it is maximized by the clustered placement scheme.

B. *Minimizing* EAFDL

From (57) and (60), it follows that

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx (l+1)^{m-l} \, \frac{(l-1)!}{(m-1)!} \prod_{e=1}^{m-l} \left(\frac{m-e}{n-e} \right)^{m-l+1-e} . \tag{69}$$

Remark 7: From (69), it follows that the placement that minimizes EAFDL does not depend on λ, b and c.

Depending on the values of m and l, we consider the following two cases:

1) $m-l=1$: For $m-l=1$, (69) yields

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx \frac{m}{n-1} \begin{cases} = 1 & \text{for } n = m+1 \\ < 1 & \text{for } n \geq m+2 . \end{cases} \tag{70}$$

2) $m-l \geq 2$: For $m-l \geq 2$, (69) can be written as follows:

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} \approx \frac{l+1}{m-1} \, \frac{l+1}{m-2} \cdots \frac{l+1}{l} \, \frac{l}{n-m+l} \prod_{e=1}^{m-l-1} \left(\frac{m-e}{n-e} \right)^{m-l+1-e} . \tag{71}$$

Using (63), (71) yields

$$\frac{\text{EAFDL}^{\text{declus}}}{\text{EAFDL}^{\text{clus}}} < \frac{l+1}{n-m+l} \leq \frac{l+1}{(m+1)-m+l} = 1 . \tag{72}$$

Remark 8: From the preceding, it follows that the EAFDL is minimized by the declustered placement scheme. In particular, when $m-l=1$ and $n=m+1$, the clustered and declustered placement schemes yield the same EAFDL.

C. *Minimizing* $E(H)$

From (58) and (61), and using (63), it follows that

$$\frac{E(H)^{\text{declus}}}{E(H)^{\text{clus}}} \approx \prod_{e=1}^{m-l} \frac{m-e}{n-e} < 1 . \tag{73}$$

Remark 9: From (73), it follows that the declustered placement minimizes $E(H)$ for any λ, b, c.

D. *Synopsis*

We summarize our findings regarding the reliability offered by the data placement schemes as follows. Independently of the device reliability characteristics and mean expressed by $1/\lambda$, the reserved rebuild bandwidth b and the device capacity c, the declustered placement scheme minimizes the expected amount of data lost when loss occurs. Also, for $m-l=1$, the clustered placement scheme maximizes the MTTDL, but the declustered placement scheme minimizes the EAFDL. However, for $m-l \geq 2$, and for practical values of n and m, the declustered placement scheme maximizes the MTTDL and at the same time minimizes the EAFDL.

V. Reliability Comparison

Here, we assess the relative reliability of the declustered placement, which according to Remarks 6, 8 and 9 is the optimal one, under various codeword lengths m. We perform a fair comparison by considering systems with the same amount of user data, U, stored under the same storage efficiency, s_{eff}. From (2), it follows that the number of devices n is fixed. Also, from (1) it follows that

$$m - l = (1 - s_{\text{eff}}) \, m = h \, m , \tag{74}$$

where h is given by

$$h \triangleq 1 - s_{\text{eff}} \tag{75}$$

and is fixed.

Using (74) to substitute l in (59) and (60) yields

$$\text{MTTDL}^{\text{declus}} \approx \frac{1}{n\,\lambda} \left[\frac{b}{[(1-h)m+1]\,\lambda\, c} \right]^{hm} (hm)! \prod_{e=1}^{hm} \left(\frac{n-e}{m-e} \right)^{hm-e} , \tag{76}$$

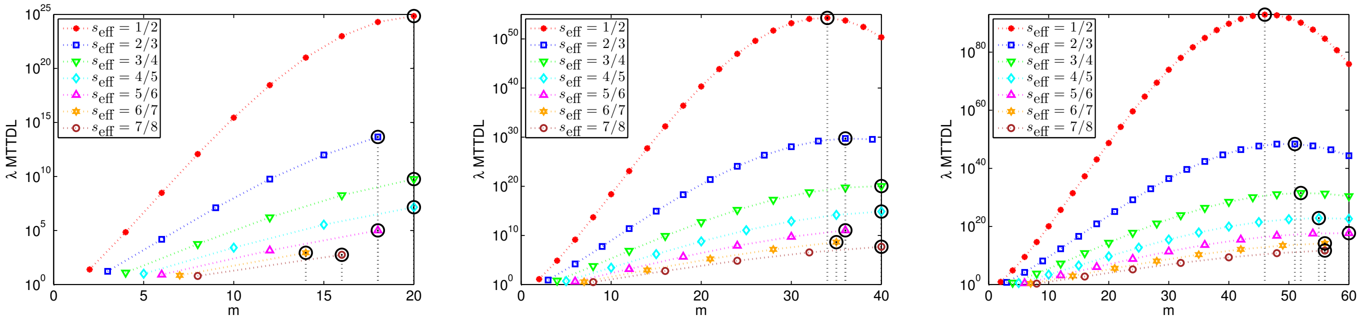
and

$$\text{EAFDL}^{\text{declus}} \approx \lambda \left[\frac{[(1-h)m+1]\,\lambda\, c}{b} \right]^{hm} \frac{m}{(hm+1)!} \prod_{e=1}^{hm} \left(\frac{m-e}{n-e} \right)^{hm+1-e} . \tag{77}$$

As discussed in Section III-A, the direct-path-approximation method yields accurate results when the storage devices are highly reliable, that is, when the ratio λ/μ of the mean rebuild time $1/\mu$ to the mean time to failure of a device $1/\lambda$ is very small. We proceed by considering systems for which it holds that $\lambda/\mu = \lambda\, c/b = 0.001$. The combined effect of the number of devices and the system efficiency on the $\lambda\,\text{MTTDL}^{\text{declus}}$ measure is obtained by (76) and shown in Figure 1 as a function of the codeword length. The values for the storage efficiency are chosen to be fractions of the form $z/(z+1)$, $z=1,\ldots,7$, such that the first point of each of the corresponding curves is associated with the single-parity $(z, z+1)$-erasure code, and the second point of each of the corresponding curves is associated with the double-parity $(2z, 2z+2)$-erasure code. We observe that the MTTDL increases as the storage efficiency s_{eff} decreases. This is because, for a given m, decreasing s_{eff} implies decreasing l, which in turn implies increasing the parity symbols $m-l$ and consequently improving MTTDL.

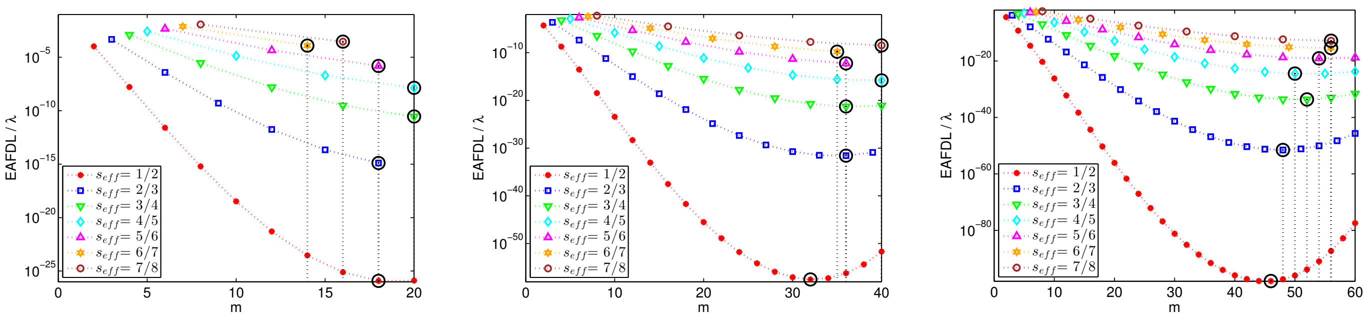
Let us now consider the single-parity codewords, which correspond to the first points of the curves. As s_{eff} increases, so do m and l, which results in a decreasing MTTDL for these codewords. This is due to the fact that as m increases, there are l data symbols, that is, more data symbols associated with each parity. This is in accordance with the results presented in Figure 2 of [26]. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.

The combined effect of the number of devices and the system efficiency on the $\text{EAFDL}^{\text{declus}}/\lambda$ measure is obtained



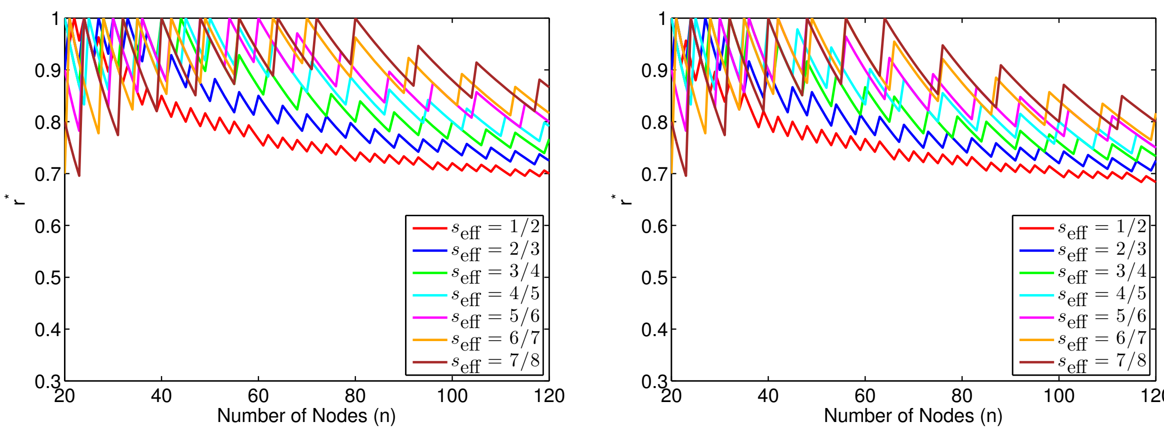
(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 1. MTTDL$^{\rm declus}$ vs. codeword length for $s_{\rm eff} = 1/2, 2/3, 3/4, 4/5, 5/6, 7/8$; $\lambda/\mu = 0.001$.



(a) $n = 20$ (b) $n = 40$ (c) $n = 60$

Figure 2. EAFDL$^{\rm declus}$ vs. codeword length for $s_{\rm eff} = 1/2, 2/3, 3/4, 4/5, 5/6, 7/8$; $\lambda/\mu = 0.001$.



(a) MTTDL (b) EAFDL

Figure 3. r^* vs. number of devices for $s_{\rm eff} = 1/2, 2/3, 3/4, 4/5, 5/6, 7/8$; $\lambda/\mu = 0.001$.

by (77) and shown in Figure 2 as a function of the codeword length. We observe that the EAFDL increases as the storage efficiency $s_{\rm eff}$ increases. Also, as $s_{\rm eff}$ increases, the EAFDL for the single-parity codewords, which correspond to the first points of the curves, also increases. We observe that the same applies for the double-parity codewords, which correspond to the second points of the curves.

We now proceed to identify the optimal codeword length, m^*, that maximizes the MTTDL or minimizes the EAFDL for a given storage efficiency. The optimal codeword length is dictated by two opposing effects on reliability. On the one hand, large values of m imply that codewords can tolerate more device failures, but on the other hand result in a higher exposure degree to failure as each of the codewords is spread across a large number of devices. In Figures 1 and 2, the optimal values, m^*, are indicated by the circles, and the corresponding codeword lengths are indicated by the vertical dotted lines. We observe that for small values of n, it holds that $m^* = n$, whereas for large values of n it holds that $m^* < n$. By comparing Figures 1 and 2, we deduce that in general the optimal codeword lengths for MTTDL and EAFDL are similar, but not identical. Let us define by r^* the ratio of m^* to n, that is

$$r^* \triangleq \frac{m^*}{n} \,. \tag{78}$$

The r^* values for various values of the system efficiency and for the two metrics of interest are shown in Figure 3. From Figures 3(a) and (b) we deduce that the optimal codeword lengths for MTTDL and EAFDL are similar, and for some values of n even identical. It can be proved that as n grows to infinity, the r^* values for MTTDL and EAFDL approach a common value that depends on s_{eff} and is roughly equal to 0.6.

VI. Conclusions

We considered the Mean Time to Data Loss (MTTDL) and the Expected Annual Fraction of Data Loss (EAFDL) reliability metrics of storage systems using advanced erasure codes. A methodology was presented for deriving the two metrics analytically. Closed-form expressions capturing the effect of various system parameters were obtained for the symmetric, clustered and declustered data placement schemes. We established that the declustered placement scheme offers superior reliability in terms of both metrics. Subsequently, a thorough comparison of the reliability achieved by the declustered placement scheme under various codeword configurations was conducted. The results obtained show that the optimal codeword lengths for MTTDL and EAFDL are similar and, as the system size grows, approach a common value that depends only on the storage efficiency.

Extending the methodology developed to derive the reliability of erasure coded systems under arbitrary rebuild time distributions and in the presence of unrecoverable latent errors is a subject of further investigation.

References

[1] D. A. Patterson, G. Gibson, and R. H. Katz, “A case for redundant arrays of inexpensive disks (RAID),” in Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 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 Comput. Surv., vol. 26, no. 2, Jun. 1994, pp. 145–185.

[3] M. Malhotra and K. S. Trivedi, “Reliability analysis of redundant arrays of inexpensive disks,” J. Parallel Distrib. Comput., vol. 17, Jan. 1993, pp. 146–151.

[4] W. A. Burkhard and J. Menon, “Disk array storage system reliability,” in Proceedings of the 23rd International Symposium on Fault-Tolerant Computing, Jun. 1993, pp. 432–441.

[5] K. S. Trivedi, Probabilistic and Statistics with Reliability, Queueing and Computer Science Applications, 2nd ed. New York: Wiley, 2002.

[6] Q. Xin, E. L. Miller, T. J. E. Schwarz, D. D. E. Long, S. A. Brandt, and W. Litwin, “Reliability mechanisms for very large storage systems,” in Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST), Apr. 2003, pp. 146–156.

[7] T. J. E. Schwarz, Q. Xin, E. L. Miller, D. D. E. Long, A. Hospodor, and S. Ng, “Disk scrubbing in large archival storage systems,” in Proceedings of the 12th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Oct. 2004, pp. 409–418.

[8] S. Ramabhadran and J. Pasquale, “Analysis of long-running replicated systems,” in Proc. 25th IEEE International Conference on Computer Communications (INFOCOM), Apr. 2006, pp. 1–9.

[9] B. Eckart, X. Chen, X. He, and S. L. Scott, “Failure prediction models for proactive fault tolerance within storage systems,” in Proceedings of the 16th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2008, pp. 1–8.

[10] K. Rao, J. L. Hafner, and R. A. Golding, “Reliability for networked storage nodes,” IEEE Trans. Dependable Secure Comput., vol. 8, no. 3, May 2011, pp. 404–418.

[11] J.-F. Pâris, T. J. E. Schwarz, A. Amer, and D. D. E. Long, “Highly reliable two-dimensional RAID arrays for archival storage,” in Proceedings of the 31st IEEE International Performance Computing and Communications Conference (IPCCC), Dec. 2012, pp. 324–331.

[12] I. Iliadis and V. Venkatesan, “An efficient method for reliability evaluation of data storage systems,” in Proceedings of the 8th International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ), Apr. 2015, pp. 6–12.

[13] ——, “Most probable paths to data loss: An efficient method for reliability evaluation of data storage systems,” International Journal on Advances in Systems and Measurements, vol. 8, no. 3&4, Dec. 2015, pp. 178–200.

[14] V. Venkatesan and I. Iliadis, “A general reliability model for data storage systems,” in Proceedings of the 9th International Conference on Quantitative Evaluation of Systems (QEST), Sep. 2012, pp. 209–219.

[15] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. Rao, “A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors,” ACM Trans. Storage, vol. 4, no. 1, May 2008, pp. 1–42.

[16] A. Thomasian and M. Blaum, “Higher reliability redundant disk arrays: Organization, operation, and coding,” ACM Trans. Storage, vol. 5, no. 3, Nov. 2009, pp. 1–59.

[17] K. M. Greenan, J. S. Plank, and J. J. Wylie, “Mean time to meaningless: MTTDL, Markov models, and storage system reliability,” in Proceedings of the USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage), Jun. 2010, pp. 1–5.

[18] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, “Disk scrubbing versus intradisk redundancy for RAID storage systems,” ACM Trans. Storage, vol. 7, no. 2, Jul. 2011, pp. 1–42.

[19] “Amazon Simple Storage Service.” [Online]. Available: http://aws.amazon.com/s3/ [retrieved: March 2017]

[20] D. Borthakur et al., “Apache Hadoop goes realtime at Facebook,” in Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 2011, pp. 1071–1080.

[21] R. J. Chansler, “Data availability and durability with the Hadoop Distributed File System,” ;login: The USENIX Association Newsletter, vol. 37, no. 1, 2013, pp. 16–22.

[22] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The Hadoop Distributed File System,” in Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies (MSST), May 2010, pp. 1–10.

[23] I. Iliadis and V. Venkatesan, “Expected annual fraction of data loss as a metric for data storage reliability,” in Proceedings of the 22nd Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Sep. 2014, pp. 375–384.

[24] C. Huang et al., “Erasure coding in Windows Azure Storage,” in Proceedings of the USENIX Annual Technical Conference (ATC), Jun. 2012, pp. 15–26.

[25] “IBM Cloud Object Storage.” [Online]. Available: www.ibm.com/cloud-computing/products/storage/object-storage/how-it-works/ [retrieved: March 2017]

[26] V. Venkatesan and I. Iliadis, “Effect of codeword placement on the reliability of erasure coded data storage systems,” in Proceedings of the 10th International Conference on Quantitative Evaluation of Systems (QEST), Sep. 2013, pp. 241–257.

[27] V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, “Reliability of clustered vs. declustered replica placement in data storage systems,” in Proceedings of the 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Jul. 2011, pp. 307–317.

[28] V. Venkatesan and I. Iliadis, “Effect of codeword placement on the reliability of erasure coded data storage systems,” IBM Research Report, RZ 3827, Aug. 2012.