RZ 3876 (#ZUR1403-054) 03/24/2014
Computer Science 10 pages

# Research Report

## Expected Annual Fraction of Data Loss as a Metric for Data Storage Reliability

Ilias Iliadis and Vinodh Venkatesan

IBM Research – Zurich
8803 Rüschlikon
Switzerland

# Expected Annual Fraction of Data Loss as a Metric for Data Storage Reliability

Ilias Iliadis and Vinodh Venkatesan
IBM Research – Zurich
8803 Rüschlikon, Switzerland

***Abstract*—Several redundancy and recovery schemes have been developed to enhance the reliability of storage systems. The effectiveness of these schemes has predominately been evaluated based on the mean time to data loss (MTTDL) metric, which has been proven useful for assessing tradeoffs, for comparing schemes, and for estimating the effect of the various parameters on system reliability. In the context of distributed and cloud storage systems, for economical reasons, it is of great importance to also consider the magnitude along with the frequency of data loss. We focus on the following reliability metric: the expected annual fraction of data loss (EAFDL), that is, the fraction of stored data that is expected to be lost by the system annually. We present a general methodology to obtain the EAFDL metric analytically, in conjunction with the MTTDL metric, for various redundancy schemes and for a large class of failure time distributions that also includes real-world distributions, such as Weibull and gamma. As a demonstration, we subsequently apply this methodology to derive these metrics analytically and to assess the reliability of a replication-based storage system under clustered, declustered, and symmetric data placement schemes. We show that the declustered placement scheme offers superior reliability in terms of both metrics. Previous work has used simulation to evaluate the magnitude of data loss, but this is the first work to analytically assess it, and the first to present a general theoretical framework for this context.**

## I. Introduction

Storage systems suffer from data losses due to component failures, including disk and node failures, as well as media failures, including unrecoverable and latent media errors. Over the years, a large variety of redundancy and recovery schemes have been developed to enhance the reliability of storage systems. The effectiveness of these schemes has predominately been assessed based on the mean time to data loss (MTTDL) metric, which is typically obtained analytically by using Markov models [1]. The results obtained are often approximate because it is often assumed that the times to component failures are independent and exponentially distributed, which does not hold in practice [2]. Recently, a methodology for obtaining MTTDL was presented under general non-exponential failure and rebuild time distributions, which therefore does not involve any Markov analysis [3]. The MTTDL metric has been proven useful for assessing tradeoffs, for comparing schemes, and for estimating the effect of various parameters on system reliability [4], [5], [6], [7].

Data losses are also encountered in the context of distributed and cloud storage systems. Therefore replication and recovery protocols are employed to cope with this issue. However, in this context, it is of great importance to consider the annual amount of data lost along with the time until data loss occurs. For example, the Reduced Redundancy Storage (RRS) option within Amazon S3 enables customers to store data at a desired level of redundancy. The RRS option stores objects on multiple devices across multiple facilities, providing 400 times the durability of a typical disk. The RRS is designed to provide 99.999999999% (eleven nines) durability and 99.99% availability of data over a given year. This durability level corresponds to an average annual expected loss of a fraction of $10^{-11}$ of the data stored in the system [8]. Such data loss events have also been documented in practice by Yahoo! [9], LinkedIn [10] and Facebook [11]. In our quest to reduce the amount of data lost, it is imperative to assess the implications of system design choices not only on the frequency of data loss events, which is captured by the MTTDL metric, but also on the amount of data lost in each loss event. Such metrics are also of importance in the context of peer-to-peer (P2P) storage systems [12].

In this work, we consider the following reliability metric: the expected annual fraction of data loss (EAFDL), that is, the fraction of stored data that is expected to be lost by the system annually. We subsequently demonstrate that this metric can be evaluated in parallel with the MTTDL. Furthermore, the EAFDL, just as the MTTDL metric, tends to be insensitive to the failure time distributions within the large class defined in [3], which also includes real-world distributions, such as Weibull and gamma. As a demonstration, we subsequently apply the methodology developed to obtain these metrics and assess the reliability of a replication-based storage system under clustered, declustered, and symmetric data placement schemes. The MTTDL and EAFDL metrics are derived analytically. The results obtained reveal that the declustered placement scheme offers superior reliability in terms of both metrics.

Note that the EAFDL metric is meant to complement, not to replace the traditional MTTDL, which yields an estimation of the frequency of data losses. These two metrics provide a useful profile of the size and frequency of data losses. Depending on the application and underlying service, some providers may prefer frequent, small losses, whereas others may prefer fewer loss events even if the loss penalty is larger [13]. For example, in the case of Facebook, each data loss event reportedly incurs an additional high fixed cost that is not proportional to the amount of data lost. It is therefore preferable to have fewer incidents of data loss with more data each than more incidents with less data [13]. Consequently, for storage systems with similar EAFDL, the most preferable system would be the one with the maximum MTTDL.

The remainder of the paper is organized as follows. Section II provides a survey of the relevant literature on reliability metrics. Section III describes the storage system model and the corresponding parameters considered. Section IV presents the general framework and methodology for deriving the EAFDL metric. Subsequently, this metric is evaluated analytically in Section V for the case of replication-based storage systems. Closed-form expressions for the clustered, declustered, and symmetric placement schemes are derived. Section VI shows the reliability of the clustered and declustered placement schemes for $r = 2, 3, 4$. Section VII provides a discussion on the use and interpretation of the EAFDL metric. Finally, we conclude in Section VIII.

## II. Related Work

The Normalized Magnitude of Data Loss (NOMDL) metric was proposed in [6]. It measures the expected amount of data lost per usable terabyte within a mission interval. It is stated that this metric could be evaluated analytically using Markov chain models, but it is argued that this approach should not be further pursued because Markov models do not accurately capture the performance of storage systems. Subsequently, it is suggested to use Monte Carlo simulation to calculate the NOMDL. In this paper, we present (and validate by means of simulation) an analytical approach that does not involve any Markov analysis, and therefore avoids the deficiencies of Markov models.

The Fraction of Data Loss Per Year (FDLPY) metric, which is equivalent to the EAFDL metric, was considered in [12]. The impact of various placement schemes on the FDLPY and MTTDL metrics was assessed by means of simulation. Furthermore, the MTTDL was estimated analytically, although the "buddy" and "global" placement schemes correspond to the clustered and declustered schemes that were evaluated analytically in [14], [15]. The present work is the first to theoretically analyze the expected annual fraction of data loss and obtain closed-form expressions that allow its dependence on the various system parameters to be assessed.

## III. Storage System Model

The storage system considered comprises $n$ storage devices (nodes or disks), with each device storing an amount $c$ of data, 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. In the remainder of the paper we present the methodology for obtaining the EAFDL for systems using data replication. Note, however, that this methodology is quite general in that it can also be directly applied to obtain the EAFDL for systems employing other redundancy schemes, such as the erasure coded system considered in [15].

TABLE I
Notation of system parameters

| Parameter | Definition |
|---|---|
| $n$ | number of storage devices |
| $c$ | amount of data stored on each device |
| $r$ | replication factor |
| $k$ | spread factor of the data placement scheme |
| $b$ | reserved rebuild bandwidth per device |
| $1/\lambda$ | mean time to failure of a storage device |
| $U$ | amount of user data stored in the system ($U = n\,c/r$) |
| $1/\mu$ | time to read an amount $c$ of data at a rate $b$ from a device ($1/\mu = c/b$) |

In a replication-based storage system, user data is replicated $r$ times with the $r$ replicas, also referred to as copies, stored in the system in such a way that no two related replicas are in the same device. The amount of user data $U$ stored in the system is then given by

$$U = \frac{n\,c}{r} \, . \tag{1}$$

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

Upon a device failure, its data have $r-1$ replicas left, which are stored in some or all of the remaining $n-1$ devices. The system subsequently rebuilds the lost copies (replicas) of this data by recovering them from the surviving devices. When the rebuild operation completes, the redundancy of this data is restored to the initial factor of $r$ replicas. A certain proportion of the device bandwidth is reserved for recovery with $b$ denoting the actual reserved rebuild bandwidth per device.

To illustrate the usefulness of the proposed EAFDL metric, we assume that the lifetimes of devices are independent and identically distributed with a mean of $1/\lambda$. An extension of the analysis to address also correlated failures is part of future work. We further consider storage devices with failure time distributions that belong to the large class defined in [3], 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 data from a device at a rebuild bandwidth of $b$, given by

$$\frac{1}{\mu} = \frac{c}{b} \, , \tag{2}$$

to the mean time to failure of a device $1/\lambda$ is small, that is, when

$$\frac{\lambda}{\mu} = \frac{\lambda\,c}{b} \ll 1 \, . \tag{3}$$

An interesting property of this class of failure time distributions is that the MTTDL reliability metric of a replication-based storage system tends to be insensitive to the distribution, that is, the MTTDL depends only on the mean value of the distribution. We will show that this also holds for the EAFDL metric.

In this work, we consider only *intelligent rebuild* schemes, that is, schemes that prioritize the rebuild of the data with the least amount of redundancy left [3], [14], [15], [16], [17], [18].

## IV. Derivation of MTTDL and EAFDL

Here we present the general methodology for deriving the EAFDL metric. It builds on the general framework for deriving the MTTDL developed in [14], [3], which we briefly review next.

At any point of time, the system can be thought to be in one of two modes: normal mode and rebuild mode. During the normal 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 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 $P_{\text{DL}}$ or back to the original normal mode by restoring all replicas, which occurs with probability $1 - P_{\text{DL}}$. Typically, the rebuild times are much shorter than the times to failure. Consequently, the time required for this complex sequence of events to complete is negligible compared with the time between successive first-device failures, and therefore can be ignored.

Let $T_i$ be the $i$th interval of a fully operational period, that is, the time interval from the time $t$ that the system is brought to its original state until a subsequent first-device failure occurs. As the system becomes stationary, the length of $T_i$ converges to $T$. In particular, for a system comprising $n$ devices with a mean time to failure of a device equal to $1/\lambda$, the expected length of $T$, is given by [3]

$$E(T) := \lim_{i \to \infty} E(T_i) = 1/(n\,\lambda) \; . \tag{4}$$

Note that the methodology presented here 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.

As each first-device failure could result in data loss with probability $P_{\text{DL}}$, the expected number of first-device failures until data loss occurs is $1/P_{\text{DL}}$. Thus, by neglecting the effect of the relatively short transient rebuild periods of the system, the MTTDL is essentially the product of the expected time between two first-device-failure events, $E(T)$, and the expected number of first-device-failure events, $1/P_{\text{DL}}$:

$$\text{MTTDL} \approx \frac{E(T)}{P_{\text{DL}}} \; . \tag{5}$$

Let $H$ denote the corresponding amount of data lost conditioned on the fact that a data loss 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:

$$\text{EAFDL} = \frac{E(H)}{\text{MTTDL} \cdot U} \; , \tag{6}$$

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 the normal operation or to 0 otherwise, that is,

$$Q = \begin{cases} H\,, & \text{if DL} \\ 0\,, & \text{if no DL}\,. \end{cases} \tag{7}$$

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

$$E(Q) = P_{\text{DL}}\, E(H) \; . \tag{8}$$

The EAFDL can alternatively be obtained as follows

$$\text{EAFDL} = \frac{E(Q)}{E(T) \cdot U} \; , \tag{9}$$

with the $E(T)$ expressed in years.

### A. Reliability Analysis

Here we demonstrate how to obtain the EAFDL analytically by extending the methodology presented in [3] for the derivation of the $P_{\text{DL}}$ and MTTDL. The EAFDL is obtained using (9), which also requires the evaluation of $E(Q)$. More specifically, $E(Q)$ is derived by considering the direct path approximation, which under condition (3) accurately assesses the reliability metrics of interest. Next we present the general outline of the methodology in more detail.

*1) Exposure Levels:* At time $t$, let $D_l(t)$ be the amount of data that has lost $l$ replicas, with $0 \le l \le r$. The system is in exposure level $e$ at time $t$, $0 \le e \le r$, if

$$e = \max_{D_l(t) > 0} l. \tag{10}$$

In other words, the system is in exposure level $e$ if there exists data with $r - e$ copies, but there is no data with fewer than $r - e$ copies in the system, that is, $D_e(t) > 0$, and $D_l(t) = 0$ for all $l > e$. At $t = 0$, $D_l(0) = 0$ for all $l > 0$ and $D_0(0)$ is the total amount of data stored in the system. Device failures and rebuild processes cause the values of $D_1(t), \cdots, D_r(t)$ to change over time, and when a data loss occurs, $D_r(t) > 0$.

*2) Direct Path to Data Loss:* Consider the direct path of successive transitions from exposure level 1 to $r$. In [14] it was shown that $P_{\text{DL}}$ can be approximated by the probability of the direct path to data loss, $P_{\text{DL,direct}}$, when devices are generally reliable, that is,

$$P_{\text{DL}} \approx P_{\text{DL,direct}} = \prod_{e=1}^{r-1} P_{e \to e+1}, \tag{11}$$

where $P_{e \to e+1}$ denotes the transition probability from exposure level $e$ to $e+1$. In fact, the above approximation holds for arbitrary device failure time distributions and the relative error tends to zero as $\lambda/\mu$ tends to zero [3].

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 loss $H$ can be approximated by that corresponding to the direct path:

$$H \approx H_{\text{direct}} . \tag{12}$$

Also, from (7) and (12) it follows that

$$Q \approx \begin{cases} H_{\text{direct}} , & \text{if DL follows the direct path} \\ 0 , & \text{otherwise} . \end{cases} \tag{13}$$

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.

*3) Amount of Data to Rebuild and Rebuild Times at Each Exposure Level:* Consider the direct path to data loss and let the amount of most-exposed data when entering exposure level $e$ be denoted by $A_e$, $e = 1, \cdots, r$. For $e < r$, $A_e$ represents the amount of data that needs to be rebuilt in that exposure level. In particular, upon the first-device failure, it holds that $A_1 = c$. Note that the amount of data lost, $H$, is the amount of most-exposed data when entering exposure level $r$, which can no longer be recovered and therefore is irrecoverably lost, that is,

$$H = A_r . \tag{14}$$

Let us also denote the rebuild times of the most-exposed data at each exposure level in this path by $R_e$ with means $E(R_e)$, $e = 1, \cdots, r-1$. Next, we will derive the conditional values for the $A_e$ and $R_e$ random variables given that the system goes through this direct path to data loss, and then we will compute the probabilities $P_{e \to e+1}$. Let $\alpha_e$ 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 [3], it was shown that, under conditions (3), $\alpha_e$ is approximately uniformly distributed between zero and one, that is,

$$\alpha_e \sim U(0,1), \quad e = 1, \cdots, r-1 . \tag{15}$$

Let $\vec{\alpha}$ denote the vector $(\alpha_1, \ldots, \alpha_{r-1})$. Note that for the assessment of MTTDL only the first $r-2$ elements are needed, whereas for the assessment of EAFDL also the last element $\alpha_{r-1}$ is required.

*4) Estimation of $P_{DL}$ and $Q$:* Consider a realization of the direct path to data loss with fractions $\alpha_e$, $e = 1, \ldots, r-1$ of the rebuild times $R_e$, $e = 1, \ldots, r-1$. Denote the vectors $(\alpha_1, \cdots, \alpha_{r-1})$ by $\vec{\alpha}$ and $(R_1, \cdots, R_{r-1})$ by $\vec{R}$. Let also $\vec{A}$ be the vector $(A_1, \cdots, A_r)$ of the most-exposed data when entering exposure levels $1, \ldots, r$. Clearly, $A_e$ depends on the values $(\alpha_1, \cdots, \alpha_{e-1})$. The length of the corresponding rebuild time $R_e$ depends on the amount of data to be recovered $A_e$, and the rebuild speed which is determined based on the data placement scheme and the rebuild bandwidth. Subsequently, a transition probability $P_{e \to e+1}$ from exposure level $e$ to $e+1$ depends on the length of the corresponding rebuild time $R_e$ and the aggregate failure rate of all devices that can potentially cause such a transition. Thus,

$$A_e = A_e(\alpha_1, \cdots, \alpha_{e-1}) = f(\alpha_1, \cdots, \alpha_{e-1}) , \tag{16}$$

$$R_e = R_e(\alpha_1, \cdots, \alpha_{e-1}) = g(\alpha_1, \cdots, \alpha_{e-1}) , \tag{17}$$

$$P_{e \to e+1} = P_{e \to e+1}(\alpha_1, \cdots, \alpha_{e-1}) = h(\alpha_1, \cdots, \alpha_{e-1}) , \tag{18}$$

for some functions $f(.)$, $g(.)$, and $h(.)$.

From (14) and (16), it follows that the amount of data lost conditioned on $\vec{\alpha}$ is then given by

$$H(\vec{\alpha}) = A_r(\vec{\alpha}) . \tag{19}$$

Also, the unconditional amount of data lost $Q$ upon a first-device failure is given by

$$Q \approx \begin{cases} H(\vec{\alpha}) , & \text{if DL under } \vec{\alpha} \\ 0 , & \text{otherwise} . \end{cases} \tag{20}$$

From (11) and (18), it follows that

$$P_{\text{DL}}(\vec{\alpha}) \approx \prod_{e=1}^{r-1} P_{e \to e+1}(\alpha_1, \cdots, \alpha_{e-1}) . \tag{21}$$

By taking the expectation of $P_{\text{DL}}(\vec{\alpha})$, (21) yields

$$P_{\text{DL}} = E[P_{\text{DL}}(\vec{\alpha})] . \tag{22}$$

By taking the expectation of $Q$, (20) yields

$$E(Q) = E[P_{\text{DL}}(\vec{\alpha})\, H(\vec{\alpha})] = E[Q(\vec{\alpha})] , \tag{23}$$

where

$$Q(\vec{\alpha}) \triangleq P_{\text{DL}}(\vec{\alpha})\, H(\vec{\alpha}) , \tag{24}$$

and by using (19)

$$Q(\vec{\alpha}) = P_{\text{DL}}(\vec{\alpha})\, A_r(\vec{\alpha}) , \tag{25}$$

The MTTDL and EAFDL are subsequently obtained based on the quantities of interest, $P_{\text{DL}}$ and $E(Q)$, via Eqs. (5) and (9), respectively.

From the preceding, it follows that the methodology developed for obtaining the MTTDL can be extended in a straightforward manner to also obtain the EAFDL metric. Next, as an application, we will show in detail the derivation of EAFDL in the context of replication-based storage systems that use clustered, declustered, and symmetric data placement schemes to protect data from node failures [3], [14], [17], [18].

## V. Replication-Based Systems

Here we consider a storage system that uses replication to protect data from device failures, in particular node failures as presented in [14]. The exact way in which the $r$ replicas of each data are stored depends on the placement scheme used. First, we consider the clustered and declustered placement schemes, as shown in Fig. 1.

*Clustered Placement (CP)*: In this placement scheme, the $n$ nodes are divided into disjoint sets of $r$ nodes and all nodes in each set are mirrors of each other, that is, they store replicas of the same data.

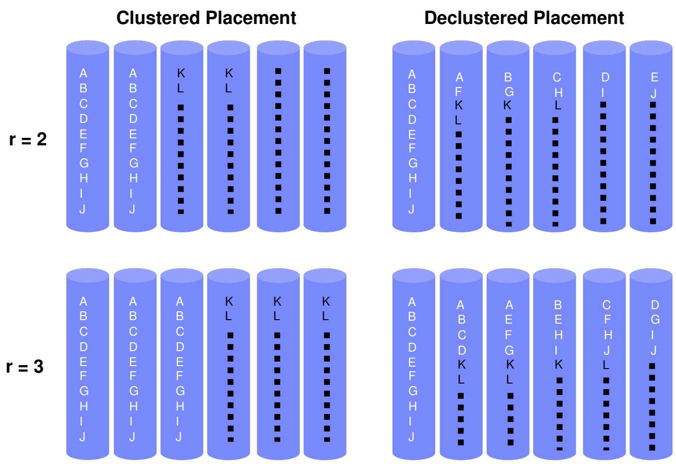


Fig. 1. In a replication-based system with $n$ devices and a replication factor of $r$, the clustered placement scheme spreads the replicas of data on each device across $r-1$ other devices, whereas declustered placement spreads the replicas of data on each device across $n-1$ other devices. An example with $n=6$ and $r=2,3$ is shown above.

*Declustered Placement (DP)*: In this placement scheme, all $\binom{n}{r}$ possible ways of placing $r$ replicas across $n$ nodes are equally used to store the data in the system. In this way, the $r-1$ replicas of the data stored on each node are equally spread across the remaining $n-1$ nodes.

These two placement schemes represent the two extremes in which the copies of the data on the failing node are spread across the remaining nodes and hence the extremes of the degree of parallelism that can be exploited when rebuilding this data. For declustered placement, the copies are spread equally across *all* remaining nodes, whereas for clustered placement, the copies are spread across the smallest possible number of nodes.

### A. Rebuild Model

As mentioned in Section III, we consider an *intelligent rebuild* scheme. This means, that in the case of two or more failures, the system always attempts to first recover the copies (replicas) of the data that has the least number of replicas left.

In declustered placement, the surviving replicas that the system has to read to recover the lost replicas are spread across all surviving nodes. Therefore, in this context, we consider a *distributed rebuild* method, in that the data to be rebuilt is read from all the nodes in which it is present, and copied to (reserved) spare space in all surviving nodes first and then to a new node. This enables a fast parallel rebuild process which reduces rebuild time and improves reliability [14], [19]. As mentioned above, during the rebuild process, a read-write bandwidth of $b$ is reserved at each node exclusively for the rebuild. In particular, as data is being read from and written to each surviving node, the total read-write rebuild bandwidth $b$ of each node is split equally between the reads and the writes, such that the effective rate of rebuild is equal to $b/2$.

For the clustered placement, when a node fails, data is read from any *one* of the surviving nodes of the cluster to which the failed node belonged and written to a spare node. Data is being read from one node and written to another using a rebuild bandwidth of $b$.

### B. Amount of Data to Rebuild at Each Exposure Level

The amount $A_e$ of the most-exposed data when entering exposure level $e$ can be derived from [14, Eqs.(7),(8)]. For the two placement schemes considered, this gives

$$A_e = \begin{cases} c \prod_{j=1}^{e-1} \alpha_j \, , & \text{for Clustered Placement (CP)} \\ c \prod_{j=1}^{e-1} \alpha_j \, \dfrac{(r-j)}{(n-j)} \, , & \text{for Declustered Placement (DP)} \end{cases}$$

$$\text{for} \quad e = 1, \ldots, r \, . \quad (26)$$

### C. Rebuild Times at Each Exposure Level

The rebuild time $R_e$ at exposure level $e$ depends on the amount of most-exposed data and the scheme used for recovering this data. In clustered placement, the amount of exposed data $A_e$ is recovered by reading this data from one node and writing it to a new spare node, such that the total rebuild bandwidth is $b$. In declustered placement, the data is read from and written to all surviving $n-e$ nodes in parallel such that the total rebuild bandwidth is $(n-e)b/2$. Consequently,

$$R_e = \begin{cases} \dfrac{A_e}{b} \, , & \text{for CP} \\ \dfrac{2\, A_e}{(n-e)\, b} \, , & \text{for DP} \end{cases} \quad \text{for} \quad e = 1, \ldots, r-1 \, . \quad (27)$$

Substituting (26) into (27) yields

$$R_e = \begin{cases} \dfrac{c}{b} \prod_{j=1}^{e-1} \alpha_j \, , & \text{for CP} \\ \dfrac{2\, c}{(n-e)\, b} \prod_{j=1}^{e-1} \alpha_j \, \dfrac{(r-j)}{(n-j)} \, , & \text{for DP} \end{cases}$$

$$\text{for} \quad e = 1, \ldots, r-1 \, . \quad (28)$$

### D. Exposure Level Transition Probability

The transition probability $P_{e \to e+1}$ from exposure level $e$ to $e+1$ depends on the duration of the corresponding rebuild time $R_e$, the number $\tilde{n}(e)$ of nodes whose failure can cause such a transition and the mean time to failure of a node $1/\lambda$. In [3, Eq.(33)] it was shown that for highly reliable storage devices, it holds that

$$P_{e \to e+1} \approx \tilde{n}(e) \, \lambda \, R_e \, , \quad (29)$$

where the relative error in the approximation goes to zero as $\lambda/\mu$ goes to zero. For clustered and declustered placement, $\tilde{n}(e)$ is equal to $r-e$ and $n-e$, respectively. Thus,

$$P_{e \to e+1} \approx \begin{cases} (r-e) \, \lambda \, R_e \, , & \text{for CP} \\ (n-e) \, \lambda \, R_e \, , & \text{for DP} \end{cases}$$

$$\text{for} \quad e = 1, \ldots, r-1 \, . \quad (30)$$

Substituting (28) into (30) yields

$$P_{e\to e+1} \approx \begin{cases} \dfrac{(r-e)\,\lambda\, c}{b} \displaystyle\prod_{j=1}^{e-1} \alpha_j \,, & \text{for CP} \\ \dfrac{2\lambda\, c}{b} \displaystyle\prod_{j=1}^{e-1} \alpha_j \, \dfrac{(r-j)}{(n-j)} \,, & \text{for DP} \end{cases}$$

$$\text{for} \quad e = 1, \ldots, r-1 \,. \quad (31)$$

*1) Estimation of $P_{DL}$ and $Q$:* Substituting (31) into (21) yields

$$P_{\text{DL}}(\vec{\alpha}) \approx \begin{cases} \left(\dfrac{\lambda c}{b}\right)^{r-1} (r-1)! \displaystyle\prod_{e=1}^{r-2} \alpha_e^{r-e-1} \,, & \text{for CP} \\ \left(\dfrac{2\lambda c}{b}\right)^{r-1} \displaystyle\prod_{e=1}^{r-2} \left(\alpha_e \, \dfrac{r-e}{n-e}\right)^{r-e-1} \,, & \text{for DP} \end{cases} \quad (32)$$

Substituting (32) and (26) into (25) yields

$$Q(\vec{\alpha}) \approx \begin{cases} \left(\dfrac{\lambda\, c}{b}\right)^{r-1} c\,(r-1)! \displaystyle\prod_{e=1}^{r-1} \alpha_e^{r-e} \,, & \text{for CP} \\ \left(\dfrac{2\lambda\, c}{b}\right)^{r-1} c \displaystyle\prod_{e=1}^{r-1} \left(\alpha_e \, \dfrac{r-e}{n-e}\right)^{r-e} \,, & \text{for DP} \end{cases} \quad (33)$$

Unconditioning (32) and (33) on $\vec{a}$ using (22) and (23), respectively, and given that its elements are independent random variables approximately distributed according to (15) such that $E(\alpha_e^k) \approx 1/(k+1)$, yields

$$P_{\text{DL}} \approx \begin{cases} \left(\dfrac{\lambda c}{b}\right)^{r-1} \,, & \text{for CP} \\ \left(\dfrac{2\lambda c}{b}\right)^{r-1} \dfrac{1}{(r-1)!} \displaystyle\prod_{e=1}^{r-2} \left(\dfrac{r-e}{n-e}\right)^{r-e-1} \,, & \text{for DP} \end{cases} \quad (34)$$

and

$$E(Q) \approx \begin{cases} \left(\dfrac{\lambda\, c}{b}\right)^{r-1} \dfrac{c}{r} \,, & \text{for CP} \\ \left(\dfrac{2\lambda\, c}{b}\right)^{r-1} \dfrac{c}{r!} \displaystyle\prod_{e=1}^{r-1} \left(\dfrac{r-e}{n-e}\right)^{r-e} \,, & \text{for DP} \end{cases} \quad (35)$$

Substituting (34) and (35) into (5) and (9), respectively, and making use of (1) and (4) yields

$$\text{MTTDL} \approx \begin{cases} \left(\dfrac{b}{\lambda\, c}\right)^{r-1} \dfrac{1}{n\,\lambda} \,, & \text{for CP} \\ \left(\dfrac{b}{2\,\lambda\, c}\right)^{r-1} \dfrac{(r-1)!}{n\,\lambda} \displaystyle\prod_{e=1}^{r-2} \left(\dfrac{n-e}{r-e}\right)^{r-e-1} \,, & \text{for DP} \end{cases} \quad (36)$$

and

$$\text{EAFDL} \approx \begin{cases} \left(\dfrac{\lambda\, c}{b}\right)^{r-1} \lambda \,, & \text{for CP} \\ \left(\dfrac{2\,\lambda\, c}{b}\right)^{r-1} \dfrac{\lambda}{(r-1)!} \displaystyle\prod_{e=1}^{r-1} \left(\dfrac{r-e}{n-e}\right)^{r-e} \,, & \text{for DP} \end{cases} \quad (37)$$

Note that the MTTDL derived in Eq. (36) for the two placement schemes is in agreement with Eqs. (17) and (20) of [14].

*Remark 1:* As $\lambda/\mu$ tends to zero, the relative errors in the approximations of MTTDL in (36) and EAFDL in (37) also tend to zero.

### *E. The Amount of Data Lost Paradox*

The metric of interest EAFDL can alternatively also be obtained via (6), which requires the derivation of $E(H)$, the expected amount of data lost. As we will demonstrate in this section, this derivation is not straightforward; on the contrary, it may easily lead to an erroneous result if it is not properly performed.

From (6) it follows that the expected amount of data lost is given by

$$E(H) = \text{MTTDL} \cdot \text{EAFDL} \cdot U \,. \quad (38)$$

Substituting (36) and (37) into (38), and recalling (1), yields

$$E(H) \approx \begin{cases} \dfrac{c}{r} \,, & \text{for CP} \\ \dfrac{c}{r} \, \dfrac{1}{\binom{n-1}{r-1}} \,, & \text{for DP} \,. \end{cases} \quad (39)$$

Alternatively, let us now consider the direct derivation of $E(H)$. From (19) and (26) it follows that

$$H(\vec{\alpha}) = \begin{cases} c \displaystyle\prod_{j=1}^{r-1} \alpha_j \,, & \text{for CP} \\ c \displaystyle\prod_{j=1}^{r-1} \alpha_j \, \dfrac{(r-j)}{(n-j)} \,, & \text{for DP} \,. \end{cases} \quad (40)$$

The expected amount $E(H)$ of data lost is obtained by unconditioning (40) on $\vec{a}$, that is $E(H) = E[H(\vec{\alpha})]$.

Considering that the elements of $\vec{a}$ are independent random variables approximately distributed according to (15) such that $E(\alpha_e) \approx 1/2$, $e = 1, \ldots, r-1$, yields for the $E(H)$

$$E(H_{\text{err}}) \approx \begin{cases} \dfrac{c}{2^{r-1}} \,, & \text{for CP} \\ \dfrac{c}{2^{r-1}} \, \dfrac{1}{\binom{n-1}{r-1}} \,, & \text{for DP} \,. \end{cases} \quad (41)$$

Comparing (39) and (41) reveals that $E(H_{\text{err}})$ is approximately equal to $E(H)$ only when $r = 2$; in all other cases $E(H_{\text{err}})$ is smaller than $E(H)$ by a factor of $2^{r-1}/r$. The explanation for this paradox is the following. For a rebuild time $R_e$, the uniform distribution of $\alpha_e$ in the interval $(0, 1)$, given by (15), holds under the assumption that there is a failure during this rebuild period. Note that no other assumptions are required to establish the uniform distribution of $a_e$, i.e., for example no assumptions on subsequent failures that may lead to data loss. However, the estimation of $E(H)$ is conditioned on the fact that data loss *does* occur, which presupposes that $r-1$ failures *do* occur during $r-1$ successive rebuilds. Under this conditioning, $a_e$ is no longer uniformly distributed. To further illustrate this issue, let us consider the time at which the second failure occurred during the first rebuild time $R_1$. If no assumptions are made on subsequent failures that may lead to data loss, according to (15), $\alpha_1$ is uniformly distributed between zero and one. Moreover, for $r > 2$ and according to (26) and (27), the duration of the second rebuild period $R_2$ is proportional to $\alpha_1$, that is, $R_2 \sim \alpha_1$, which implies that $R_2$ is also uniformly distributed in the interval $(0, R_2(\max))$. However, conditioning on the fact that an additional third failure occurs during the rebuild time $R_2$, it is more likely that the $R_2$ period is long rather than short. This implies that, under this conditioning, neither $R_2$ nor $\alpha_1$ is uniformly distributed. In this context, only the distribution of $\alpha_{r-1}$ is uniform in the interval $(0, 1)$; all other distributions of $\alpha_e$, $e = 1, \ldots, r-2$, are not. Therefore, the derivation of $E(H_{\text{err}})$ in (41) is incorrect; it underestimates the actual value because it incorrectly considers the short rebuild periods to be equally likely as the long ones.

### *F. Symmetric Schemes*

Here we consider the symmetric placement schemes that lie between the clustered and declustered schemes. For each node in the system, let its *redundancy spread factor* $k$ denote the number of nodes over which the data on that node and its corresponding redundant data are spread [17]. In a symmetric placement scheme, the $r-1$ replicas of the data on each node are *equally* spread across $k-1$ other nodes, the $r-2$ replicas of the data shared by any two nodes are equally spread across $k-2$ other nodes, and so on. According to this scheme, the system is effectively divided into $n/k$ disjoint groups of $k$ nodes. Each group contains an amount of $U/k$ user data along with all of the corresponding replicas that are placed in its $k$ nodes in a declustered manner. Clearly, the clustered and declustered placement schemes are special cases of symmetric placement schemes in which $k$ is equal to $r$ and $n$, respectively.

Let us denote by $\text{MTTDL}_k$ and $\text{EAFDL}_k$ the metrics corresponding to a group. Clearly, $\text{MTTDL}_k$ is obtained from (36) by replacing $n$ with $k$. The $\text{MTTDL}(k)$ of the system, which comprises $n/k$ groups, is given by $\text{MTTDL}(k) = \text{MTTDL}_k/(n/k)$. Thus,

$$\text{MTTDL}(k) \approx \begin{cases} \left(\dfrac{b}{\lambda\, c}\right)^{r-1} \dfrac{1}{n\,\lambda}\,, & \text{for } k = r \\ \left(\dfrac{b}{2\,\lambda\, c}\right)^{r-1} \dfrac{(r-1)!}{n\,\lambda} \displaystyle\prod_{e=1}^{r-2} \left(\dfrac{k-e}{r-e}\right)^{r-e-1}, & \text{for } r < k \le n\,. \end{cases} \tag{42}$$

Similarly, $\text{EAFDL}_k$ is obtained from (37) by replacing $n$ with $k$. The $\text{EAFDL}(k)$ of the system is equal to that of a group, that is, $\text{EAFDL}(k) = \text{EAFDL}_k)$. Thus,

$$\text{EAFDL}(k) \approx \begin{cases} \left(\dfrac{\lambda\, c}{b}\right)^{r-1} \lambda\,, & \text{for } k = r \\ \left(\dfrac{2\,\lambda\, c}{b}\right)^{r-1} \dfrac{\lambda}{(r-1)!} \displaystyle\prod_{e=1}^{r-1} \left(\dfrac{r-e}{k-e}\right)^{r-e}, & \text{for } r < k \le n\,. \end{cases} \tag{43}$$

TABLE II
RANGE OF VALUES OF DIFFERENT SIMULATION PARAMETERS

| Parameter | Definition | Range |
|---|---|---|
| $n$ | number of storage nodes | 4 to 64 |
| $c$ | amount of data stored on each node | 12 TB |
| $r$ | replication factor | $2, 3, 4$ |
| $b$ | reserved rebuild bandwidth per node | 96 MB/s |
| $1/\lambda$ | mean time to failure of a node | 350 to $10^4$ h |
| $U$ | amount of user data stored in the system | 24 to 192 TB |
| $1/\mu$ | time to read an amount $c$ of data at a rate $b$ from a node | 34.7 h |

*Remark 2:* From (42) and (43) it follows that, for $r > 2$ and $n > 4$, $\text{MTTDL}(k)$ is increasing in $k$, and $\text{EAFDL}(k)$ is decreasing in $k$. Consequently, within the class of symmetric placement schemes considered, the $\text{MTTDL}(k)$ is maximized and the $\text{EAFDL}(k)$ minimized when $k = n$. Therefore, the maximum MTTDL and the minimum EAFDL are achieved by the declustered placement scheme.

## VI. Numerical Results

Here we assess the reliability of the clustered and declustered schemes in terms of the MTTDL and EAFDL metrics. We accomplish this using both theoretical predictions and event-driven simulations. From (36) and (37), we obtain closed-form expressions for the MTTDL and EAFDL metrics for $r = 2, 3, 4$. The MTTDL expressions were initially derived in [3] for the large class of failure time distributions considered and are included in this paper for completeness.

Typical parameter values were assumed for the simulations, the same as in [3], which are listed in Table II. The parameter $1/\lambda$ was chosen to be in the range of $350$ h to $10,000$ h, which yields a $\lambda/\mu$ ratio in the range of $0.1$ to $0.0034$. The value of $10,000$ h is of the same order as that found in Google storage clusters where nodes were observed to become unavailable with an MTBF of 4.3 months, with roughly 10% of these events requiring a rebuild as they last more than 15 minutes [16]. As discussed in Section III, the theoretical analysis holds for $\lambda/\mu$ ratios satisfying Eq. (3). Accordingly, we observed that the simulation results match well with the theoretical predictions for lower $\lambda/\mu$ values. Here we show the results for $\lambda/\mu = 0.0034$ for replication factors 2 and 3, and for $\lambda/\mu = 0.1$ for replication factor 4. As expected, the simulation results for the former case match with the theoretical predictions, whereas for the latter case they deviate. Despite the deviation in the latter case, the theoretical estimates still lie within the same order of magnitude as the simulation results, and accurately capture the effect of the system size on the reliability metrics considered. According to Remark 1, and as noted in [3], if for a given value of $\lambda/\mu$ the simulation results match well with the theoretical results, they should also match well for all lower values of $\lambda/\mu$. In conjunction with the observations made for $\lambda/\mu = 0.0034$, this implies that the theoretical predictions will also be accurate for all values $\lambda/\mu < 0.0034$.

To demonstrate the validity of the theoretical model for non-exponential failure time distributions, the simulations use Weibull distributions with shape parameters greater than one.

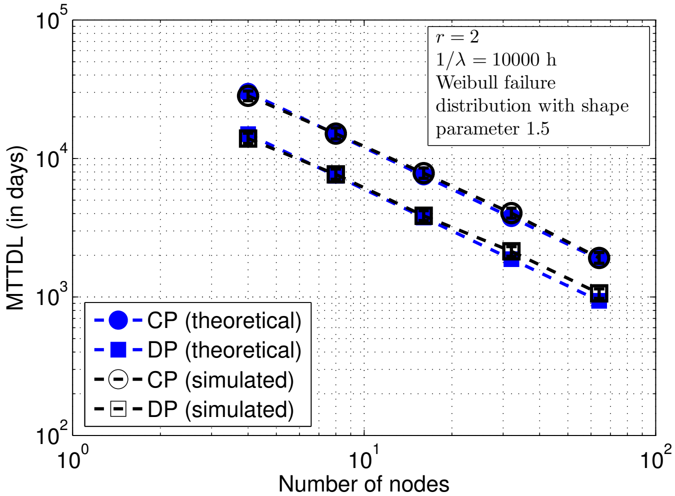


Fig. 2. MTTDL vs. number of nodes for a replication factor of two.

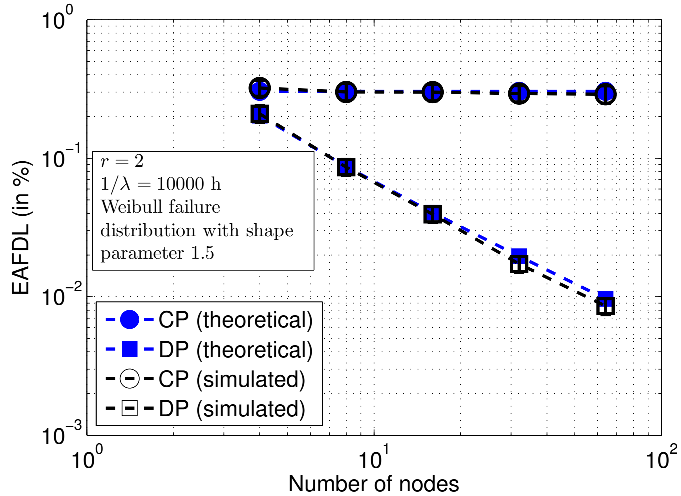


Fig. 3. EAFDL vs. number of nodes for a replication factor of two.

In contrast to an exponential distribution, which implies a constant failure rate over time, a Weibull distribution with a shape parameter greater than one indicates increasing failure rates over time. This is a reasonable model for lifetimes of realistic nodes. However, as predicted by theory, the choice of shape parameter, and hence the choice of failure time distribution does not significantly affect the results. Therefore, as an example we show the results only for a shape parameter equal to 1.5. For each set of parameters, the simulation is run at least 100 times and the MTTDL and EAFDL values, along with their 95% confidence intervals are estimated. The confidence intervals are not clearly visible because their length is about the height of the symbols that are used to show the mean values.

**Replication Factor 2:**

$$\text{MTTDL} \approx \begin{cases} b/(n\,c\,\lambda^2)\,, & \text{for CP} \\ b/(2\,n\,c\,\lambda^2)\,, & \text{for DP}\,, \end{cases} \tag{44}$$

and

$$\text{EAFDL} \approx \begin{cases} \lambda^2\,c/b\,, & \text{for CP} \\ 2\,\lambda^2\,c/[(n-1)\,b]\,, & \text{for DP}\,. \end{cases} \tag{45}$$

Figs. 2 and 3 illustrate that the theoretical results closely match the simulation ones for a small number of nodes. For a large number of nodes, for example $n = 64$, simulation results slightly deviate from the theoretical ones. This is due to the fact that $\lambda/\mu = 0.0034$ is no longer small enough for the approximations to hold. However, we observed that if the failure time distribution were exponential, then there would be no deviation in this range. Both the clustered and the declustered placement schemes have an MTTDL that is inversely proportional to the number of nodes, with the declustered placement having a slightly worse MTTDL (by a factor of two) than the clustered placement. In contrast, the EAFDL for the clustered placement scheme is proportional to the number of nodes, whereas the EAFDL for the declustered placement scheme is essentially independent of the number of nodes. Consequently, the declustered placement scheme offers a higher reliability than the clustered one.

**Replication Factor 3:**

$$\text{MTTDL} \approx \begin{cases} b^2/(n\,c^2\,\lambda^3)\,, & \text{for CP} \\ (n-1)\,b^2/(4\,n\,c^2\,\lambda^3)\,, & \text{for DP}\,, \end{cases} \tag{46}$$

and

$$\text{EAFDL} \approx \begin{cases} \lambda^3\,c^2/b^2\,, & \text{for CP} \\ 8\,\lambda^3\,c^2/[(n-1)^2\,(n-2)\,b^2]\,, & \text{for DP}\,. \end{cases} \tag{47}$$

Figs. 4 and 5 illustrate that the theoretical results closely match the simulation ones. As seen from the above equations, the MTTDL of clustered placement is inversely proportional to the number of nodes, whereas the MTTDL of declustered placement is essentially independent of the number of nodes. The EAFDL of clustered placement scheme is proportional to the number of nodes, whereas the EAFDL of declustered placement is essentially inversely proportional to the square of the number of nodes. Consequently, the declustered placement scheme offers greatly superior reliability.

**Replication Factor 4:**

$$\text{MTTDL} \approx \begin{cases} b^3/(n\,c^3\,\lambda^4)\,, & \text{for CP} \\ (n-1)^2\,(n-2)\,b^3/(24\,n\,c^3\,\lambda^4)\,, & \text{for DP}\,, \end{cases} \tag{48}$$

$$\text{EAFDL} \approx \begin{cases} \lambda^4\,c^3/b^3\,, & \text{for CP} \\ 48\,\lambda^4\,c^3/[(n-1)^3\,(n-2)^2\,(n-1)\,b^3]\,, & \text{for DP}\,. \end{cases} \tag{49}$$

To ensure that the simulation running times are not prohibitively high, the mean time to failure of a node is chosen to be equal to 350 h. In this case, it holds that $\lambda/\mu = 34.7/350 \approx 0.1$, which may not be small enough for the approximations

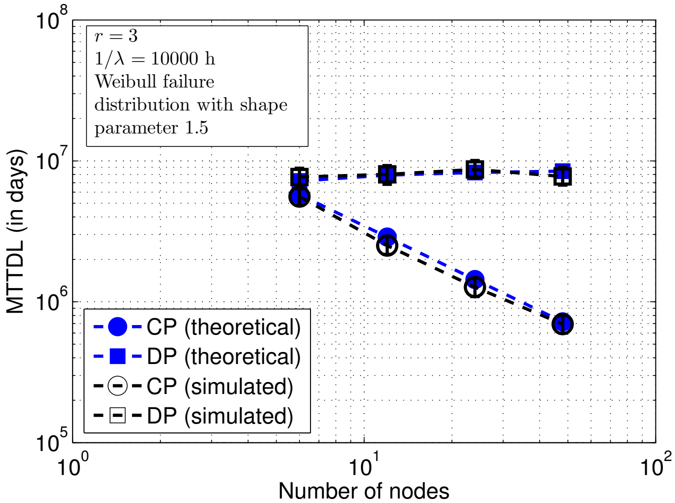


Fig. 4. MTTDL vs. number of nodes for a replication factor of three.

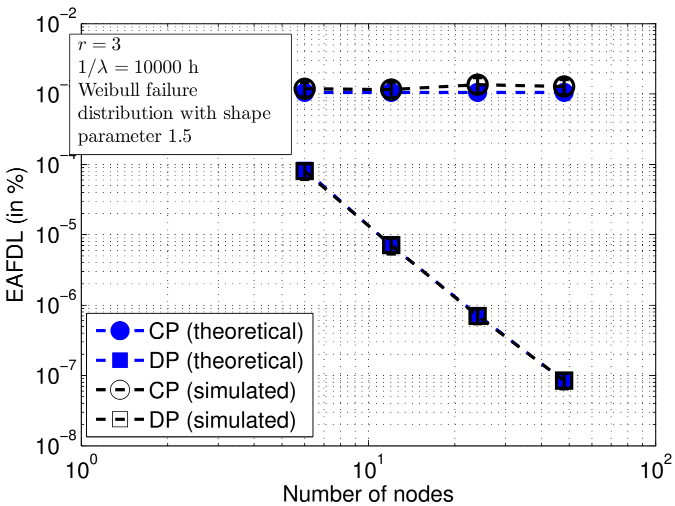


Fig. 5. EAFDL vs. number of nodes for a replication factor of three.

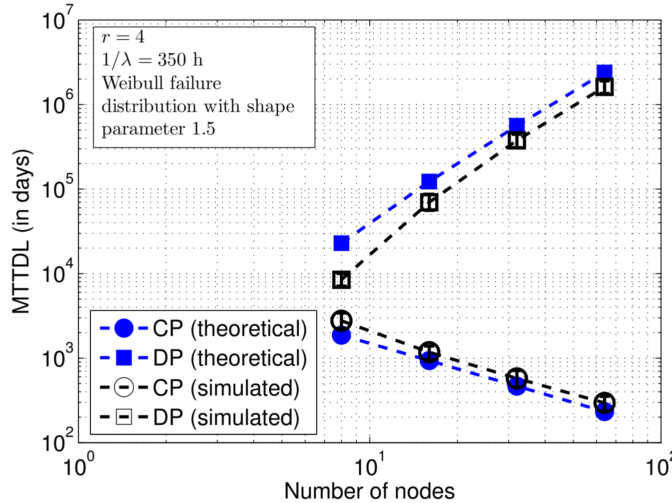


Fig. 6. MTTDL vs. number of nodes for a replication factor of four.

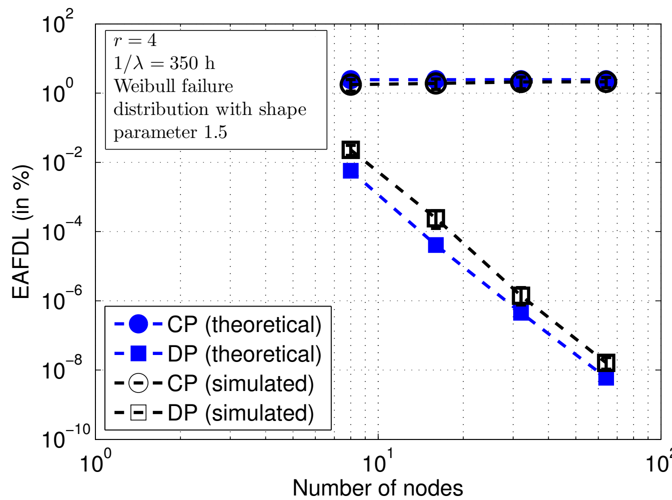


Fig. 7. EAFDL vs. number of nodes for a replication factor of four.

of the theoretical analysis to hold. As expected, Figs. 6 and 7 show that the simulation-based MTTDL and EAFDL results are slightly different than the corresponding theoretical ones. Nevertheless, for both metrics of interest, the slopes of the lines in the figures predicted by theory match well with those reflected from the simulation results. As seen from the above equations, the MTTDL of clustered placement is inversely proportional to the number of nodes, whereas the MTTDL of declustered placement increases essentially proportionally to the square of the number of nodes. The EAFDL of clustered placement scheme is proportional to the number of nodes, whereas the EAFDL of declustered placement is essentially inversely proportional to the fifth power of the number of nodes. Consequently, the declustered placement scheme again offers vastly superior reliability.

## VII. Discussion

The expected annual fraction of data loss (EAFDL) metric was introduced to provide an assessment of the amount of data lost and has to be used cautiously. Suppose, for example, that EAFDL is equal to $10^{-3}$. This does not necessarily imply that $0.1\%$ of the user data is lost each year. To see why, consider two different storage systems with the same EAFDL, equal to $10^{-3}$, but different MTTDLs, namely 10 years and 100 years. The system with an MTTDL of 10 years is expected to have more frequent data loss events than the other one. However, according to (6), upon a data loss event, the former system is expected to lose only $1\%$ $(= E(H)/U)$ of the data, whereas the latter is expected to lose $10\%$ of the data. Note that both MTTDL and EAFDL are expectations of stochastic random variables, and therefore the actual time to data loss and the amount of data lost may be very different from their respective expectations.

The desired reliability profile of a system, that is, the desired values of MTTDL and EAFDL, depends on the application and underlying service. If, for example, losing an order of $10\%$ of the data in a loss event is unacceptable, then only the former storage system satisfies this requirement. In general, consider a requirement that the fraction of stored data that is expected to be lost by the system in a loss event should not exceed a value of $f$. Noting that this fraction is given by $E(H)/U$, and according to (6), the following relation should be satisfied

$$\text{EAFDL} \cdot \text{MTTDL} \leq f . \tag{50}$$

In another scenario, consider a system comprised of $N$ independent and identical storage subsystems storing an amount $U$ of user data each, with reliability characteristics specified by MTTDL and EAFDL, and operating in parallel. The total amount of user data $U_{\text{sys}}$ stored in the system is then given by

$$U_{\text{sys}} = N\, U . \tag{51}$$

The MTTDL of the system, denoted by $\text{MTTDL}_{\text{sys}}$, is approximately given by

$$\text{MTTDL}_{\text{sys}} \approx \frac{\text{MTTDL}}{N} . \tag{52}$$

When a data loss occurs in the system, the expected amount of data lost, $E(H_{\text{sys}})$, is equal to the expected amount of data lost in the corresponding subsystem, that is,

$$E(H_{\text{sys}}) = E(H) . \tag{53}$$

According to (6), the EAFDL of the system, denoted by $\text{EAFDL}_{\text{sys}}$, is then given by

$$\text{EAFDL}_{\text{sys}} = \frac{E(H_{\text{sys}})}{\text{MTTDL}_{\text{sys}} \cdot U_{\text{sys}}} . \tag{54}$$

Substituting (51), (52), and (53) into (54), and making use of (6), yields

$$\text{EAFDL}_{\text{sys}} \approx \frac{E(H)}{\text{MTTDL} \cdot U} = \text{EAFDL} . \tag{55}$$

Consequently, the EAFDL value of the system is approximately equal to that of the subsystems. Furthermore, if $N$ is large, the MTTDL is small, and therefore user data may be lost every year, with the EAFDL expressing the fraction of stored data that is expected to be lost by the system annually.

## VIII. Conclusions

We considered the expected annual fraction of data loss (EAFDL) metric, which assesses the reliability level achieved in the context of distributed and cloud storage systems. This metric, together with the traditional MTTDL metric, provide a useful profile of the size and frequency of data losses. Our work is the first to assess the magnitude of data loss analytically and demonstrate that the EAFDL and MTTDL metrics can be evaluated in parallel in a common general theoretical framework. We derived the EAFDL by extending the general methodology developed to obtain the MTTDL of systems using various redundancy schemes and for a large class of failure time distributions that also includes the real-world distributions, such as Weibull and gamma. We subsequently applied this methodology to derive the amount of data lost in the case of replication-based storage systems that use clustered and declustered data placement schemes. The MTTDL and EAFDL metrics were obtained analytically in closed-form expressions. The results obtained show that the declustered placement scheme offers superior reliability in terms of both metrics.

Application of the methodology developed to derive the EAFDL for systems using other redundancy schemes, such as erasure codes, is a subject of future work.

## References


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

[2] B. Schroeder and G. A. Gibson, “Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?” in *Proc. 5th USENIX Conference on File and Storage Technologies (FAST)*, Feb. 2007, pp. 1–16.

[3] 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)*, Sep. 2012, pp. 209–219.

[4] 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, pp. 1–42, 2008.

[5] 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.

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

[7] 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, pp. 1–42, 2011.

[8] “Amazon Simple Storage Service,” 2013. [Online]. Available: http://aws.amazon.com/s3/

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

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

[11] D. Borthakur *et al.*, “Apache Hadoop goes realtime at Facebook,” in *Proc. ACM SIGMOD Int’l Conference on Management of Data*, Jun. 2011, pp. 1071–1080.

[12] S. Caron, F. Giroire, D. Mazauric, J. Monteiro, and S. Pérennes, “P2P storage systems: Study of different placement policies,” *Peer-to-Peer Networking and Applications*, pp. 1–17, Mar. 2013.

[13] A. Cidon *et al.*, “Copysets: Reducing the frequency of data loss in cloud storage,” in *Proc. USENIX Annual Technical Conference (ATC)*, Jun. 2013, pp. 37–48.

[14] 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 Int’l Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Jul. 2011, pp. 307–317.

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

[16] D. Ford *et al.*, “Availability in globally distributed storage systems,” in *Proc. 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Oct. 2010, pp. 61–74.

[17] 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 Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Aug. 2012, pp. 189–197.

[18] V. Venkatesan and I. Iliadis, “Effect of latent errors on the reliability of data storage systems,” in *Proc. 21th Annual IEEE Int’l Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Aug. 2013, pp. 293–297.

[19] Q. Xin, E. L. Miller, and T. J. E. Schwarz, “Evaluation of distributed recovery in large-scale storage systems,” in *Proc. 13th IEEE Int’l Symposium on High Performance Distributed Computing (HPDC)*, Jun. 2004, pp. 172–181.