Effect of Replica Placement on the Reliability of Large-Scale Data Storage Systems

Vinodh Venkatesan, Ilias Iliadis, Xiao-Yu Hu, Robert Haas
IBM Research - Zurich
{ven, ili, xhu, rha}@zurich.ibm.com

Christina Fragouli
École Polytechnique Fédérale de Lausanne
christina.fragouli@epfl.ch

Abstract—Replication is a widely used method to protect large-scale data storage systems from data loss when storage nodes fail. It is well known that the placement of replicas of the different data blocks across the nodes affects the time to rebuild. Several systems described in the literature are designed based on the premise that minimizing the rebuild times maximizes the system reliability. Our results however indicate that the reliability is essentially unaffected by the replica placement scheme. We show that, for a replication factor of two, all possible placement schemes have mean times to data loss (MTTDLs) within a factor of two for practical values of the failure rate, storage capacity, and rebuild bandwidth of a storage node. The theoretical results are confirmed by means of event-driven simulation. For higher replication factors, an analytical derivation of MTTDL becomes intractable for a general placement scheme. We therefore use one of the alternate measures of reliability that have been proposed in the literature, namely, the probability of data loss during rebuild in the critical mode of the system. Whereas for a replication factor of two this measure can be directly translated into MTTDL, it is only speculative of the MTTDL behavior for higher replication factors. This measure of reliability is shown to lie within a factor of two for all possible placement schemes and any replication factor. We also show that for any replication factor, the clustered placement scheme has the lowest probability of data loss during rebuild in critical mode among all possible placement schemes, whereas the declustered placement scheme has the highest probability. Simulation results reveal however that these properties do not hold for the corresponding MTTDLs for a replication factor greater than two. This indicates that some alternate measures of reliability may not be appropriate for comparing the MTTDL of different placement schemes.

I. INTRODUCTION

In today's large-scale distributed storage systems, vast amounts of user data are stored among a large number of nodes and disks. Distributed peer-to-peer storage systems, such as Farsite, OceanStore, CFS, PAST, Glacier, and Shark, aim at providing inexpensive, highly-available storage without centralized servers (see [1] and the references therein). In the presence of component failures, such as node and disk failures, reliability, long-term durability, and high availability are ensured by storing user data in a redundant manner. Redundancy is achieved by employing the established, widely used replication and erasure coding schemes.

Large-scale data storage systems use various redundancy schemes to prevent data loss that can occur because of multiple node failures. Replication is one of the widely used schemes where each data block is replicated and the replicas are stored in different nodes to improve the chances that at least one replica survives when multiple storage nodes fail. To maintain redundancy in the system, whenever a node fails, a rebuild process is initiated to create copies of the blocks that were lost. Wide-scale replication increases the reliability, availability, and durability, but it also increases the bandwidth and storage requirements of the system.

How the replicas are placed plays an important role in how much time the rebuild process takes, and this in turn affects the reliability of the system. In this paper, upper and lower bounds are derived on one particular measure of reliability of the storage system for all possible replica placement schemes. To keep the problem analytically tractable, the measure of reliability that is used is different from the usual measure of reliability, the mean time to data loss (MTTDL). For a replication factor of two, this measure of reliability allows an explicit calculation of MTTDL, whereas for higher replication factors, it is only speculative of the nature of the MTTDL.

The theoretical results obtained strongly indicate that this measure of reliability is affected only negligibly by the choice of the replica placement scheme for a wide range of node failure rates and node rebuild rates. This is further supported by event-driven simulations which agree with the theoretical MTTDL predictions for a replication factor of two. However, for a replication factor of three, simulation results show that this is no longer true, and that the reliability will be strongly affected by the choice of replica placement scheme. This implies that the measure of reliability used, while suitable for predicting the MTTDL for a replication factor of two, is no longer suitable for predicting the MTTDL for higher replication factors. From our simulation results, we conjecture that another measure of reliability may be more appropriate for predicting the MTTDL behavior. As a result, we have also considered an alternate measure of reliability and verified its appropriateness.

For a replication factor of two, we demonstrate that reducing rebuild times, and consequently the *window of vulnerability*, does not necessarily lead to improved reliability. This is because the reliability depends not only on the window of vulnerability, but also on the number of nodes that have the replicas of the data in the node lost. Distributing replicas across many nodes increases the probability that a second failure affects some of these replicas, thereby causing data loss.

The remainder of the paper is organized as follows: Section II discusses some related work; Section III describes

the storage system model and the parameters considered; Section IV describes two measures of reliability of a storage system; Section V contains the main contribution of this paper; Section VI gives the derivation of the main result; Section VII shows event-driven simulation results on MTTDL to compare with the theoretical predictions; and Section VIII concludes the paper.

II. RELATED WORK

Data placement issue has been considered in [2]. The emphasis of that work was on redundancy placement, namely, the placement of erasure coded data, rather than replica placement. The reliability of a system with the number of nodes equal to the replication factor is addressed in [3]. The paper provides an explicit expression of MTTDL for such a system.

Decentralized storage systems, such as CFS, OceanStore, Ivy, and Glacier, use replication to provide reliability, but employ a variety of different strategies for placement and maintenance. In architectures that employ distributed hash tables (DHTs), the choice of algorithm for data replication and maintenance can have a significant impact on both performance and reliability [4]. The paper proposes five different placement schemes. The scheme that minimizes the probability of data loss is the *block placement* scheme, in which replicated data is stored in the same set of nodes. Similar results are also presented in [5], [6]. The findings of these works match with our *theoretical* results, which also show that less distributed schemes have higher reliability.

System reliability depends both on the recovery mechanism and on the replica placement scheme. Fast recovery schemes reduce the window of vulnerability and therefore improve the system reliability [7], [8], [9]. Rebuild times are reduced by appropriate replica placement strategies. In particular, distributing replicas over many storage nodes in the system aids in quick rebuild upon failure. However, their analysis is based on an idealistic assumption that replica-sets (referred to as redundancy sets and groups in [7], [8], and as objects in [9]) fail independently. In contrast, in our analysis we assume that nodes fail independently and take into account the correlations among different replica-sets that this induces. As we show in this paper, this leads to different results.

Largely two approaches have been taken in comparing reliabilities of systems with different placement schemes: (i) approximate methods to compute MTTDL [9], and (ii) use of measures of reliability other than MTTDL, such as the probability that a storage system survives without data loss until time t as a function of t [5], [6], and probability of data loss within a fixed period of time [8], [4]. In this paper, we take the latter approach and use simulations to compare the MTTDL behavior to the behavior of the measure used.

III. SYSTEM MODEL

The model and assumptions of the storage system considered and the failure and rebuild model used are described in this section. Table I lists the different parameters used.

A. Storage System

The storage system considered is a block-based storage system comprising n storage nodes with total data storage capacity of nc bytes, where c is the capacity of each storage node. Every user data block is of size s bytes, and is replicated r times. These r replicas are stored in the system such that no two replicas of a data block are in the same node. The exact way in which the r replicas of each data block are stored depends on the placement scheme used. For our theoretical calculations, we impose no restriction on the set of placement schemes that can be used. However, for simulations, we use the following three schemes: *(a) declustered*, *(b) clustered*, and *(c) k-clustered placement*.

(a) Declustered Placement: The r replicas of each data block are stored in some r nodes out of the n nodes in the system. There are $\binom{n}{r}$ ways of choosing r nodes from the n nodes. In this placement scheme, all $\binom{n}{r}$ choices are equally used for storing replicas. Therefore, when a node fails, the replicas of the blocks in the failed node will be spread over all remaining nodes. As the total capacity of the system is nc and the total size of a block and its replicas is rs, this placement is possible only if $s \leq nc/\left(r\binom{n}{r}\right)$. We typically consider a node with capacity $c = 12$ TB (consisting of 12 disks, each having 1 TB capacity [10]) and number of nodes from $n = 4$ to $n = 96$ in our simulations. For declustered placement to be possible for 100 nodes, the size of a data block s must be less than about 121.2 GB and 2.5 GB for a replication factor of two and three, respectively. Furthermore, for all $\binom{n}{r}$ choices to be equally used, the number of data blocks must be a multiple of $\binom{n}{r}$. If it is not, there will be differences in the number of blocks that are shared among different sets of nodes. However, for realistic values of node numbers, e.g. $n \leq 1000$, and small blocks sizes, e.g. $s \leq 10$ MB, this difference is negligible.

(b) Clustered Placement: In this placement scheme, the n nodes are divided into disjoint sets of r nodes. All r nodes in a given set are mirrors of each other, that is, they store replicas of the same set of data blocks.

(c) k-Clustered Placement: This is a generalization of above two schemes. In this placement scheme, the n nodes are divided into disjoint sets of k nodes called *clusters*. Each of these clusters is an independent storage system with k nodes with a declustered placement scheme. No data block in one cluster is replicated in another cluster. It is easy to see that n-clustered placement is the same as declustered placement and r-clustered placement is the same as clustered placement. So for different values of k between r and n, we have a broad range of different placements schemes.

B. Failure Model

Storage nodes are comprised of one or more disks, a memory, processor, network interface, and power supply. Typically, these components are less reliable than the disks, and the failure of any of these components leads to a node failure [11]. The disks inside a node are assumed to be protected by a RAID scheme which also corrects unrecoverable or latent sector errors by using either scrubbing or intra-disk redundancy [12]. Disk failures are assumed to cause a node failure only if the RAID system is unable to recover from these failures. Furthermore, in systems that use SMART (Self-Monitoring Analysis and Reporting Technology) for disks, the disks can be aggressively replaced to ensure that disk failures are not the main cause of node failures. Therefore, in our model, node failures are assumed to be caused primarily by the failure of components other than disks, such as the RAID-controller, memory, processor, network interface, and power supply. The failure of these components, and therefore the failure of nodes, is assumed to be independent with exponentially distributed times to failure. In particular, the time to failure of a node, T_F, is assumed to be exponentially distribute with rate λ, that is, $T_F \sim \exp(\lambda)$. Note that this assumption is in contrast to disk failures, which are neither independent nor exponentially distributed [13], [14]. However, the above model may not apply to node failures that are caused by software bugs, DDoS attacks, virus/worm infections, node overloads and human error, as these factors may result in correlated node failures [15].

C. Rebuild Model

When a node fails, a rebuild process is initiated to restore the lost replicas and bring the system back to its original state, in which each block has r replicas. Spare space is assumed to be reserved on each node for rebuild, and the new replicas of the lost blocks are created in the spare space of the surviving nodes. Once the new replicas have been created in the spare space, a new node is brought in and these newly created replicas are transferred to the new node. The main advantage of creating replicas in the spare space first as opposed to creating replicas directly in a new node is that rebuild can be done in parallel (distributed rebuild, see Fig. 1) using the rebuild bandwidth available at many surviving nodes, thereby reducing the rebuild time. Also, once the replicas have been created, the system can survive another node failure without data loss. If a new node was brought in first and the new replicas were created directly in the new node, the rebuild speed would be restricted to the write bandwidth available at the new node. This leads to a higher probability that any of the surviving nodes fail before rebuild completes, thereby causing data loss.

During the rebuild process, an average read-write bandwidth of b bytes/s is assumed to be available at each node for rebuild. This is usually only a fraction of the total bandwidth available at each node, the remainder being used to serve system-user requests. During the rebuild process, let there be w_i data blocks that are to be read from node i and w_i' blocks that are
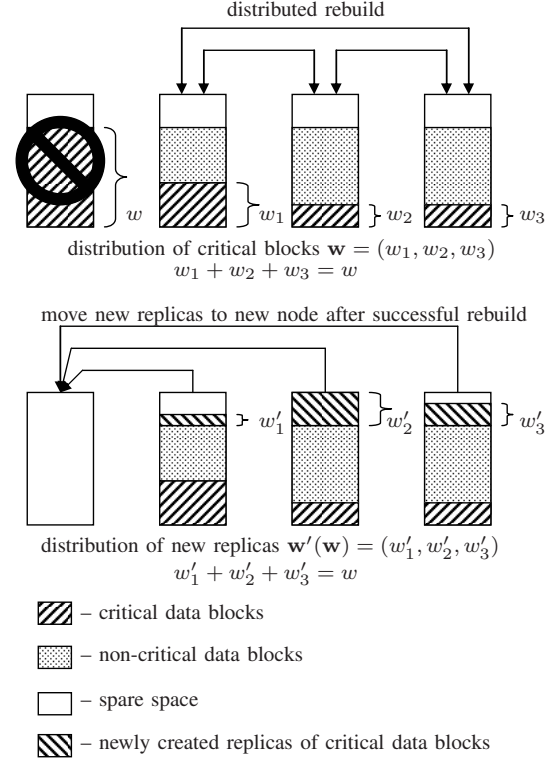

Fig. 1. Example of rebuild model for a replication factor $r = 2$ system. When one node fails, the critical data blocks are present in the surviving nodes. The rebuild process creates replicas of these critical blocks by copying them from one surviving node to another in parallel.

to be written to node i. The *effective* bandwidth b_i available for *reading* blocks from node i is proportional to the amount of data read, that is,

$$b_i = \frac{w_i}{w_i + w_i'} b. \tag{1}$$

One may argue that the fastest way to read out all w_i blocks before node i fails would be to first read all these blocks using the full rebuild bandwidth b available, and then to write the w_i' blocks. However, in a large system, if all $\sum_i w_i = w$ blocks from all the nodes are read in parallel at full rebuild bandwidth without copying them to other nodes simultaneously, then a large reliable buffer capable of storing data at rates of up to $(n-1)b$ is required to store these w blocks until they have been fully read out, and then copy them from the buffer to the nodes. We assume that in general such a large buffer capable of storing data at such high speeds is not available. This means that we need to interleave the reads and writes at each node, that is, a few blocks are read out from each node, then the copies of these blocks are written to other nodes, and so on until all blocks have been read and copied. This results in an effective read bandwidth b_i as given above.

We assume that a typical disk has a read-write bandwidth of 40 MB/s. Therefore a node with 12 disks will have about 480 MB/s of read-write bandwidth. During the rebuild process, if a fifth of this is used for rebuild on an average, then $b = 96$ MB/s. For a system with $n = 100$ nodes, the network

bandwidth must support up to about $(n-1)b \approx 9.6$ GB/s for exchange of data among all nodes during rebuild. This number is, however, the worst-case estimate, which holds for the case of declustered placement because the replicas of the lost data blocks are present in all surviving nodes. On the other hand, clustered placement only requires about $rb = 0.2$ and 0.3 GB/s of network bandwidth for exchange of data during rebuild for a replication factor of 2 and 3, respectively. This is because in clustered placement the replicas of the lost data block are present only in $(r-1)$ nodes, and new replicas are written "effectively" to one node. In this paper, the network bandwidth is considered to be sufficiently high ($> (n-1)b$) to allow exchange of data among all nodes during rebuild.

The mean time to complete *reading* all w_i blocks from node i is given by

$$E\left[T_R^i(w_i, w_i')\right] = \frac{w_i s}{b_i} = \begin{cases} \dfrac{s}{b}(w_i + w_i') & \text{if } w_i \neq 0, \\ 0 & \text{if } w_i = 0. \end{cases} \quad (2)$$

The nodes are expected to serve user requests while performing rebuild and there is also some randomness in the location of the data to be rebuilt. As the bandwidth b available for rebuild is only on *average*, the time to complete reading all w_i blocks at node i is assumed to be exponentially distributed, that is,

$$T_R^i(w_i, w_i') \sim \exp\left(1/E\left[T_R^i(w_i, w_i')\right]\right). \quad (3)$$

It is further assumed that $T_R^i(w_i, w_i')$ is independent of $T_R^j(w_j, w_j')$ for $i \neq j$ and independent of the times to failure T_F^i. This is because, once w_i and w_i' are fixed for all nodes, the only sources of randomness are the location of the blocks to be read and the serving of user requests. The location of these blocks and the user requests are not assumed to have any specific patterns that might induce correlations in the access times across different nodes.

IV. MEASURES OF RELIABILITY

Two measures of reliability that will be used in this paper are described in this section.

A. Mean Time to Data Loss (MTTDL)

A data loss is said to have occurred in the system if the replicas of at least one data block have been lost by the system and cannot be restored. The average time it takes for the system until a data loss event occurs, also referred to as the mean time to data loss (MTTDL), is a well-known measure of reliability of the system that is widely used by the storage systems community.

Analytically computing this measure for a replication-based system with a given replica placement scheme under certain failure and rebuild models of the nodes is intractable except for a few select cases, such as for the basic mirroring scheme for replications factors $r \geq 2$ [3]. To circumvent this problem, some authors have proposed approximate continuous-time Markov chain models that enable analytical tractability of the MTTDL computation [9], whereas others have proposed

different measures of reliability, such as the probability that a storage system survives without data loss until time t as a function of t [5], [6]. We take the latter approach and use a different measure of reliability, namely, the probability of data loss during rebuild in the critical mode of the system. This measure has also been used in [16, Eq. (38)]. The difference to earlier work [5], [6] is that in this measure we also take the rebuild time into account. Rebuild time, and hence the time window of vulnerability, is known to be greatly affected by the placement scheme used and hence it is an important factor to be considered in measuring reliability. For a replication factor of two, we directly relate the newly introduced measure to the MTTDL.

B. Probability of Data Loss during Rebuild in Critical Mode

To keep the problem analytically tractable, a simple measure of reliability is used. Assume that at a given point in time, $(r-1)$ nodes of the system, chosen uniformly at random, have failed. This will result in the loss of one or more replicas of some user data blocks. Data blocks that have lost $(r-1)$ copies and have only one other copy surviving in the system are called *critical blocks*. Data blocks that have 2 or more copies in the system are called *non-critical blocks*. The nodes containing these critical blocks are called *critical nodes* and the system is said to be in a *critical mode* when there is at least one critical block in the system. The rebuild process attempts to first create replicas of these critical blocks to prevent data loss that can occur if any of the critical nodes fail. The measure of reliability $\mathscr{P}$ is defined as

$$\mathscr{P} = \Pr\{\text{DL}|(r-1) \text{ nodes failed}\}, \quad (4)$$

where DL is the event that data loss occurs because of a critical node failure before the critical blocks in that node have been copied to another node. If this event does not occur, the system goes to a new state upon exiting the critical mode. As the rebuild of non-critical data blocks is still pending, this state is different from the initial one with all data blocks having r replicas. Modeling of the exact operation of the system using a Markov chain requires an enormous number of such intermediate states. This is why the evaluation of MTTDL is intractable.

The probability of loss of non-critical data blocks caused by two or more node failures before the rebuild of critical blocks completes is typically of higher order than $\mathscr{P}$ and hence ignored. This can be seen in two placement examples: (i) In declustered placement, if a node fails after the critical blocks in it have been copied to another node, it does not result in data loss. However, if two or more nodes fail after the corresponding critical data in them have been copied to other nodes, it could result in the loss of non-critical data blocks. The probability of such an event happening before the rebuild completes is however negligible compared to $\mathscr{P}$. (ii) In clustered placement, each mirrored set is an independent storage system which is unaffected by node failures and rebuilds in other mirrored sets. Given that $(r-1)$ nodes failed, there are two cases: (a) all these $(r-1)$ nodes belonged to

the same mirrored set, or (b) all these $(r-1)$ nodes did not belong to the same mirrored set. In case (a), the system is in critical mode, but all the non-critical data have r replicas. This means that the probability of losing a non-critical data block by losing r nodes in the same mirrored set before the rebuild in the critical mirrored set completes is negligible compared to $\mathscr{P}$. In case (b), there are no critical blocks to rebuild and so the time to "rebuild" critical blocks is zero. Therefore, the probability of losing non-critical data blocks before rebuild completes is zero.

For a replication factor of two, the measure of reliability $\mathscr{P}$ can be directly translated to MTTDL when the mean time to rebuild is much smaller than the mean time to failure, that is, when $E[T_R] \ll E[T_F]$.

$$\text{MTTDL} = \frac{1}{n\lambda\mathscr{P}}, \quad r=2, \ E[T_R] \ll E[T_F]. \quad (5)$$

This can be shown for a k-clustered placement (and thereby for declustered and clustered placements as well) as follows. Let $\text{MTTDL}_{\text{clus.}}$ be the MTTDL of a cluster. As the clusters are independent of each other and there are n/k clusters in total, the MTTDL of the system is $\text{MTTDL}_{\text{clus.}}/(n/k)$. In a given cluster, the loss of a node leads to critical mode. The mean time taken to lose a node is $E[T_F] = E[\min\{T_F^1, \cdots, T_F^k\}] = \frac{1}{k\lambda}$. As the probability of data loss in critical mode is $\mathscr{P}$, the cluster enters critical mode $\frac{1}{\mathscr{P}}$ times on average before data loss occurs. Assuming that the time to rebuild $E[T_R] \ll E[T_F]$, $\text{MTTDL}_{\text{clus.}} = \frac{1}{k\lambda} \times \frac{1}{\mathscr{P}}$. Therefore, MTTDL $= \text{MTTDL}_{\text{clus.}}/(n/k) = \frac{1}{n\lambda\mathscr{P}}$.

V. Effect of Replica Placement on Reliability

The following proposition shows that the measure of reliability $\mathscr{P}$ is not affected much (within a factor of two) by the choice of replica placement scheme.

Proposition V.1. *For the system model described in Section III, for all possible replica placement schemes and for any replication factor $r \geq 2$, the measure of reliability $\mathscr{P}$ as defined in Section IV-B is bounded as follows:*

$$\frac{\lambda nc}{b} \frac{1}{\binom{n}{r-1}} \frac{1}{\left(1+\frac{\lambda c}{b}\right)} \leq \mathscr{P} < \frac{2\lambda nc}{b} \frac{1}{\binom{n}{r-1}}. \quad (6)$$

Proof: The proof is given in Section VI. ∎

These bounds lie approximately within a factor of two for all practical values of the failure rate λ, node rebuild bandwidth b, and node capacity c. For $\lambda = 10^{-5}\,\text{h}^{-1}$, $b = 480$ MB/s, and $c = 12$ TB, the factor $\left(1+\frac{\lambda c}{b}\right)$ is equal to 1.001 which implies that the reliability of all placement schemes for a given replication factor is of about the same order. Even for low-power systems such as FAWN [17] and Pergamum [18] with comparable failure rate λ and an order of magnitude higher rebuild time $\frac{c}{b}$, the factor $\left(1+\frac{\lambda c}{b}\right)$ is equal to 1.01 and the bounds are still close to each other.

Furthermore, as will be shown in Section VI-B, Lemma VI.1, the upper bound corresponds to the case of uniformly distributed placement of replicas, also referred to as random placement [9]; and the lower bound corresponds to the case

of mirrored placement of replicas, also referred to as basic mirroring [5]. Similar results have also been obtained by others [5], [6], albeit only for a select number of schemes and a replication factor of two. The above result differs from the already known results in two main ways: (1) it holds for *all possible placement schemes* and for *any replication factor*, and (2) it takes the effect of rebuild process into account as it is known that the rebuild times can differ vastly for different placement schemes. The intuition behind the result is that, in critical mode, when all n critical nodes take part in rebuild in parallel, the rebuild time can be reduced n times; however, as the failure of *any* of these n nodes during rebuild can result in data loss, the probability that data loss occurs during rebuild in critical mode stays the same. The factor two stems from the fact that, in declustered placement, each node does an equal number of reads and writes of critical data, thereby reducing the *effective* rebuild bandwidth per node to $b/2$, whereas in clustered placement, the nodes having critical data only do reads and the rebuild bandwidth is b. Therefore, under this measure of reliability, clustered placement is about a factor two better than declustered placement. We formalize this intuition in the above Proposition.

For a replication factor of two, we have the following bound on MTTDL:

$$\frac{b}{2\lambda^2 nc} < \text{MTTDL} \leq \frac{b}{\lambda^2 nc}\left(1+\frac{\lambda c}{b}\right), \quad r=2. \quad (7)$$

The above bound follows from (5) and Proposition V.1. Once again, we observe that, for practical values of λ, c/n, and b, the MTTDL lies within a factor of two for all placement schemes. The clustered placement scheme has the highest MTTDL, which is about a factor of two better than that of declustered placement scheme. This is validated by event-driven simulation results in Section VII.

Remark 1. Note that the bounds on $\mathscr{P}$ (and correspondingly on MTTDL for a replication factor of two) do not depend on the size of data blocks s. Although large-sized data blocks may not permit certain placement schemes, the bounds still hold true for all s, $0 < s \leq c$.

VI. Proof of Proposition V.1

The definition of $\mathscr{P}$ in (4) can be expanded as follows: when $(r-1)$ nodes fail, let there be w critical blocks. Denote the distribution of critical blocks in the surviving nodes by $\mathbf{w} = (w_1, \cdots, w_{(n-r+1)})$, where w_i is the number of critical blocks in the ith surviving node, and $\sum_{i=1}^{n-r+1} w_i = w$. As the $(r-1)$ failed nodes were chosen uniformly at random from the n nodes in the system, the replica placement scheme chosen induces a probability distribution on $\mathbf{w}$, $\Pr\{\mathbf{w}|(r-1) \text{ nodes failed}\}$. In the critical mode, the rebuild process attempts to make replicas of these critical blocks before the failure of a critical node results in data loss. Let $\mathbf{w}'(\mathbf{w}) = (w_1', \cdots, w_{(n-r+1)}')$ denote the distribution of the first replicas of all the critical blocks in the surviving nodes. Once these replicas are created, the system is no longer in critical mode. As this distribution is chosen such that replica of

a critical block from a node is not created in the same node, the distribution $\mathbf{w}'$ depends on $\mathbf{w}$ itself, and therefore is expressed as a function of $\mathbf{w}$. Note that there may be more than one choice of $\mathbf{w}'$ for a given $\mathbf{w}$, that is, $\mathbf{w}'(\mathbf{w})$ is not unique. When computing the bounds on $\mathscr{P}$, we find the choices of $\mathbf{w}'(\mathbf{w})$ that maximize and minimize $\mathscr{P}$. Depending on the failure rate λ, the rebuild bandwidth b, the distribution of critical blocks to be read $\mathbf{w}$, the distribution of replicas of critical blocks to be written $\mathbf{w}'(\mathbf{w})$, there is a certain probability, $\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'\}$, that there is data loss because of a failure that occurs before successful rebuild of these critical blocks. The probability $\mathscr{P}$ is expressed in terms of these two conditional probabilities as follows:

$$\mathscr{P} = \sum_{\mathbf{w}} \big(\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$$
$$\times \Pr\{\mathbf{w}|(r-1) \text{ nodes failed}\}\big), \quad (8)$$

where the summation is over all possible distributions w of critical blocks under the replica placement scheme chosen.

An upper bound on $\mathscr{P}$ is obtained as follows:

$$\mathscr{P} = \sum_{w} \sum_{\mathbf{w}:\sum w_i = w} \big(\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$$
$$\times \Pr\{\mathbf{w}|(r-1) \text{ nodes failed}\}\big) \quad (9)$$
$$\leq \sum_{w} \Big(\max_{\mathbf{w}:\sum w_i = w} \max_{\mathbf{w}'(\mathbf{w})} \Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$$
$$\times \sum_{\mathbf{w}:\sum w_i = w} \Pr\{\mathbf{w}|(r-1) \text{ nodes failed}\}\Big) \quad (10)$$
$$= \sum_{w} \Big(\max_{\mathbf{w}:\sum w_i = w} \max_{\mathbf{w}'(\mathbf{w})} \Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$$
$$\times \underbrace{\Pr\{w \text{ critical blocks}|(r-1) \text{ nodes failed}\}}_{=:q(w)}\Big) \quad (11)$$
$$= \sum_{w} \Big(q(w) \max_{\mathbf{w}:\sum w_i = w} \max_{\mathbf{w}'(\mathbf{w})} \Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}\Big), \quad (12)$$

where (9) follows by splitting the sum in (8) into two parts by introducing the number of critical blocks w; (10) follows by pulling the maximum value of $\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$ out of the inner summation; and (11) follows by noting that the second summation is equivalent to $\Pr\{w \text{ critical blocks}|(r-1) \text{ nodes failed}\}$ because it counts all possible distributions of w critical blocks.

Similarly a lower bound on $\mathscr{P}$ is obtained as follows:

$$\mathscr{P} \geq \sum_{w} \Big(q(w) \min_{\mathbf{w}:\sum w_i = w} \min_{\mathbf{w}'(\mathbf{w})} \Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}\Big). \quad (13)$$

We now compute the terms inside the summation in inequalities (12) and (13) in the next three subsections.

A. Rebuild at Any One Node

Owing to our assumptions on failure and rebuild models, and by making use of (2), the probability that all the critical blocks in node i are successfully read before the node fails is given by

$$\Pr\{T_R^i(w_i, w_i') < T_F^i\}$$
$$= \frac{1}{1 + \lambda E\left[T_R^i(w_i, w_i')\right]} \quad (14)$$
$$= \begin{cases} \dfrac{1}{1 + \frac{\lambda s}{b}(w_i + w_i')} & \text{if } w_i \neq 0, \\ 1 & \text{if } w_i = 0. \end{cases} \quad (15)$$

B. Rebuild at All Nodes

In a critical mode with w critical blocks, the probability of data loss is equal to one minus the probability that each of the $(n-r+1)$ nodes successfully completes reading its critical blocks, that is,

$$\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\}$$
$$= 1 - \prod_{i=1}^{(n-r+1)} \Pr\{T_R^i(w_i, w_i') < T_F^i\}. \quad (16)$$

Substituting (15) in (16), we get

$$\Pr\{\text{DL}|\mathbf{w}, \mathbf{w}'(\mathbf{w})\} = 1 - \prod_{i \in I(\mathbf{w})} \frac{1}{1 + \frac{\lambda s}{b}(w_i + w_i')}, \quad (17)$$

where $I(\mathbf{w}) = \{i : w_i \neq 0, 1 \leq i \leq (n-r+1)\}$ is the set of critical nodes. The following lemma gives the maximum and the minimum of the above probability:

Lemma VI.1. *For any distribution of critical blocks $\mathbf{w} = (w_1, \cdots, w_{(n-r+1)})$ such that the total number of critical blocks is w, and any distribution $\mathbf{w}'(\mathbf{w}) = (w_1', \cdots, w_{(n-r+1)}')$ of the first replicas of these critical blocks such that no two replicas of the same block lie on the same node, the probability of data loss before successful completion of rebuild is bounded as follows:*

$$\frac{\lambda s w}{b} \frac{1}{\left(1 + \frac{\lambda s w}{b}\right)} \leq \Pr\{DL|\mathbf{w}, \mathbf{w}'(\mathbf{w})\} < \frac{2\lambda s w}{b}.$$

The lower bound is achieved for $\{\mathbf{w}, \mathbf{w}'(\mathbf{w})|w_j = w \text{ for some } j \text{ and } w_i = 0 \,\forall i \neq j\}$; the set $\{\mathbf{w}, \mathbf{w}'(\mathbf{w})|w_i + w_i' = \frac{2w}{n-r+1} \,\forall i\}$ achieves the highest probability of data loss.

Proof: See Appendix A. ∎

The above lemma is a key result in the proof of Proposition V.1. It shows the main points of this paper: (i) the probability of data loss during rebuild in critical mode for a given number of critical blocks w lies within a tight range of values; (ii) the lowest probability of data loss occurs when all w critical blocks are in one node, which is the case in clustered placement; and (iii) the highest probability of data loss occurs when the rebuild is uniformly distributed across all nodes, which is the case in declustered placement.

C. Expected Number of Critical Blocks

As $q(w)$, defined in (11), is the probability of having w critical blocks when $(r-1)$ nodes fail, the expected number

of critical blocks when $(r-1)$ nodes fail is given by $E[w] = \sum_w wq(w)$.

Lemma VI.2. *For any placement scheme, the expected number of critical blocks given $(r-1)$ node failures is $E[w] = \sum_w wq(w) = \frac{nc}{s\binom{n}{r-1}}$.*

Proof: Let $x_1, x_2, \cdots, x_d$ be the $d := \frac{nc}{sr}$ distinct data blocks which have been replicated r times and stored in the system. Any given x_i is replicated and stored in r different nodes. So the data block x_i can become critical when any $(r-1)$ out of these r nodes fail, which can occur in $\binom{r}{r-1} = r$ ways. Given the failure of $(r-1)$ nodes chosen uniformly at random from n nodes, the probability that x_i becomes critical is independent of the replica placement scheme and is always equal to

$$\Pr\{x_i \text{ is critical}|(r-1) \text{ nodes fail}\} = \frac{r}{\binom{n}{r-1}}, \forall i.$$

Given that there are a total of d distinct blocks, the expected number of critical blocks when $(r-1)$ nodes fail is given by

$$E[w] = d \times \Pr\{x_i \text{ is critical}|(r-1) \text{ nodes fail}\}$$
$$= \frac{nc}{sr} \times \frac{r}{\binom{n}{r-1}} = \frac{nc}{s\binom{n}{r-1}}. \qquad \blacksquare$$

D. Upper Bound

The upper bound on $\mathscr{P}$ in (6) is obtained as follows:

$$\mathscr{P} < \sum_w \frac{2\lambda sw}{b} \times q(w) = \frac{2\lambda nc}{b} \frac{1}{\binom{n}{r-1}},$$

where the inequality in the first step follows by applying Lemma VI.1 in (12), and the second step follows from Lemma VI.2.

E. Lower Bound

The lower bound on $\mathscr{P}$ in (6) is obtained as follows:

$$\mathscr{P} \geq \sum_w \frac{\lambda sw}{b} \frac{1}{(1 + \frac{\lambda sw}{b})} q(w) \qquad (18)$$

$$\geq \frac{\lambda s}{b} \frac{1}{(1 + \frac{\lambda c}{b})} \sum_w wq(w) \qquad (19)$$

$$= \frac{\lambda nc}{b} \frac{1}{(1 + \frac{\lambda c}{b})} \frac{1}{\binom{n}{r-1}}, \qquad (20)$$

where (18) follows by applying Lemma VI.1 in (13); (19) follows by noting that the number of critical blocks w cannot be greater than the number of blocks on one node, that is, $w \leq \frac{c}{s}$; and (20) follows from Lemma VI.2.

VII. SIMULATION RESULTS

A. Placement Schemes

Three different placements schemes were used in the simulations - *(a) declustered, (b) clustered,* and *(c) k-clustered placement*. From our theoretical result on MTTDL for a replication factor of two (7), we expect clustered placement to have the highest MTTDL, followed by k-clustered placement and then by declustered placement. We also expect that clustered placement is better than declustered placement by about a factor of two. This is attributed to the fact that, in declustered placement, each node performs equal number of reads and writes of critical data, thereby reducing the effective rebuild bandwidth per node to $b/2$, whereas in clustered placement, the nodes having the critical data perform reads only and therefore the rebuild bandwidth is b.

B. Simulation Method

Event-driven simulations were used to calculate the MTTDL for the three placements schemes. Three types of events drive the simulation time forward: *(a) failure events, (b) rebuild-complete events,* and *(c) node-restore events*. The state of the system is maintained by three variables - `time`, the simulated time, `activeNodes`, the number of active nodes in the system, and a vector of length $(r+1)$ `dataExposure` $= (d_0, \cdots, d_r)$, where d_i is the number of distinct data blocks that have lost i replicas. Data loss occurs when $d_r > 0$. At each event these variables are updated.

(a) Failure Event: A failure event triggers the following: (i) decreasing `activeNodes` by one, (ii) scheduling the next failure event after time $T_F(\text{activeNodes} \times \lambda)$, (iii) updating `dataExposure` by taking into account the fact that a partial rebuild of the most exposed data has occurred, and (iv) scheduling the rebuild-complete event based on the most exposed data in `dataExposure` and the placement scheme used. By nature of the rebuild process, data placement is preserved, that is, declustered remains declustered and clustered remains clustered. This is because, when the placement is declustered, critical blocks are read from and written to all nodes at the same time and the new replicas are placed such that declustering is preserved. When the placement is clustered, the replicas are created in a new node directly instead of creating them in the spare space of existing nodes first and then copying them to a new node. This preserves clustered placement. We have another tunable parameter, namely, the time taken to detect the failure of a node and start the rebuild process, T_{delay}. This is added to T_R while scheduling the rebuild-complete events. This parameter is seen to have an influence only when T_{delay} is comparable to $1/(n\lambda)$. For practical systems, $1/\lambda$ is on the order of $100,000$ h. If the system has $n = 100$ nodes, $1/(n\lambda) = 1000$ h. Typically T_{delay} is much smaller than 1000 h and so we do not present the effect of this parameter in the simulation results presented in this paper.

(b) Rebuild-Complete Event: A rebuild-complete event triggers the following (i) updating `dataExposure` by setting the amount of most exposed data to zero and adding this amount to a lower exposure level (this means that the rebuild process always creates replicas of the most exposed data first), and (ii) scheduling the node-restore event when all data have r copies (completion of rebuild process). The node-restore event is the time when all the replicas that were newly created have been successfully transferred to new nodes and the number of nodes is brought back to n. The number of nodes to restore is stored in `nodesToRestore`.

Parameter	Meaning	Range
c	storage capacity of each node	12 TB
n	number of storage nodes	4 to 100
r	replication factor	$2, 3$
b	rebuild bandwidth available at each node	96 MB/s
λ	failure rate of a storage node	$10^{-3} - 10^{-5}\,\mathrm{h}^{-1}$

(c) Node-Restore Event: This event increases `activeNodes` by `nodesToRestore`.

For each set of parameters, the simulation is run 100 times, and the MTTDL and its bootstrap 95% confidence intervals are computed. Whereas for declustered placement, the simulation is run for n nodes, for clustered and k-clustered placement, the simulations are run only for one cluster, that is, r and k nodes respectively, and the obtained MTTDL of the cluster is divided by n/r and n/k, respectively, to obtain the MTTDL of the system. This is because clusters are independent of the each other, and the number of clusters is n/r and n/k for clustered and k-clustered placement, respectively.

C. Simulation Results

Table II shows the range of different parameters that were used for the simulations. Typical values for practical systems are used for all parameters, except for the mean time to failure of a node for a replication factor of three. For simulating a system with a replication factor of three, the mean time to failure has been chosen artificially low (1000 h instead of $100,000$ h) to run the simulations fast because the running times of simulations with $\lambda = 10^{-5}\,\mathrm{h}^{-1}$ are prohibitively high. Although this approach scales down the MTTDL by making failure events more frequent, it has been used (as in [9]) because it preserves the ratios of MTTDLs of the various schemes.

Replication Factor Two: Fig. 2 shows the comparison of theoretically predicted MTTDLs (from the bounds in (7)) and simulated values of MTTDL as a function of the number of nodes for a system with a replication factor of two with declustered and clustered schemes. It is seen that the theoretical predictions are quite accurate. In addition, a third theoretical curve for the mirrored placement scheme based on the formula from [16, Eq. (46)] is plotted. It is observed that this curve coincides with the upper bound of (7), which corresponds to clustered placement. The simulated MTTDL values for 4-clustered placement scheme are found to lie between the corresponding MTTDL values for clustered and declustered placement schemes. This is in agreement with our theoretical prediction.

Replication Factor Three: Fig. 3 shows the simulated values of MTTDLs for a replication factor of three for clustered and declustered placements. Declustered placement appears to be generally better than clustered placement. The theoretical values for clustered placement based on [3, Eq. (2)] agree with the simulation values.

To investigate why the behavior of MTTDL is different from that of $\mathscr{P}$, we plot $\mathscr{P}$ obtained from the simulations
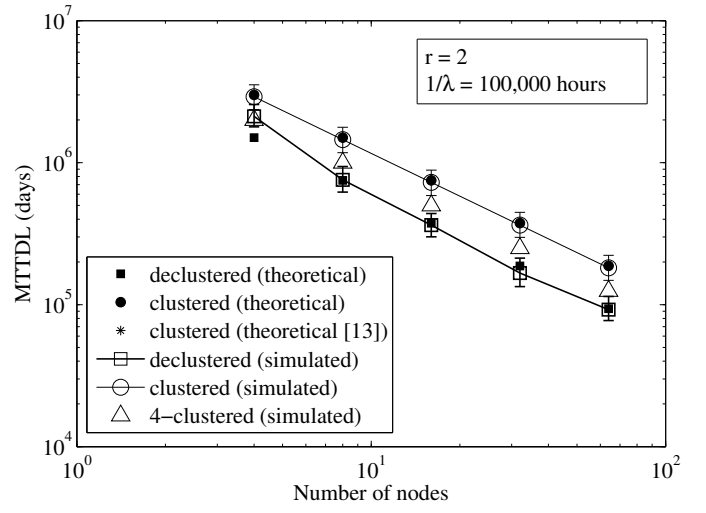

Fig. 2. Comparison of theoretically predicted and simulated values of MTTDL for a replication factor of two with mean time to failure of a node equal to $100,000$ h; For the simulated results, 95% bootstrap confidence intervals are shown.
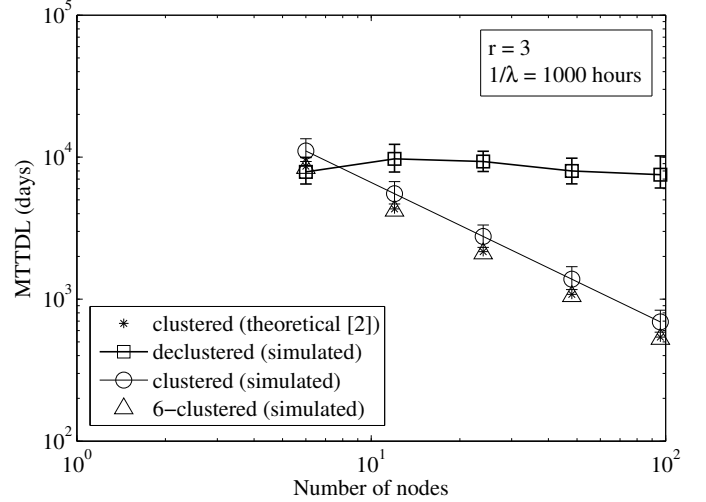

Fig. 3. Comparison of theoretically predicted and simulated values of MTTDL for replication factor three with mean time to failure of a node equal to 1000 h; For the simulated results, 95% bootstrap confidence intervals are shown.

in Fig. 4. We observe that the simulation values are half of the theoretical ones. This is because the theoretical results are obtained assuming there is no rebuild between node failures, whereas in simulation, when the second node fails, approximately half of the data in the first lost node has already been rebuilt. The simulation results, however, support the theoretical results of Proposition V.1 and Lemma VI.1 in that the declustered placement has a higher probability of data loss (by factor of two) than the clustered placement. This shows that the measure of reliability used, while being suitable for predicting the MTTDL for a replication factor of two, is no longer suitable for predicting the MTTDL for higher replication factors.

In Fig. 5 we plotted the probability of data loss given *one* node failure obtained from our simulations. This measure, the
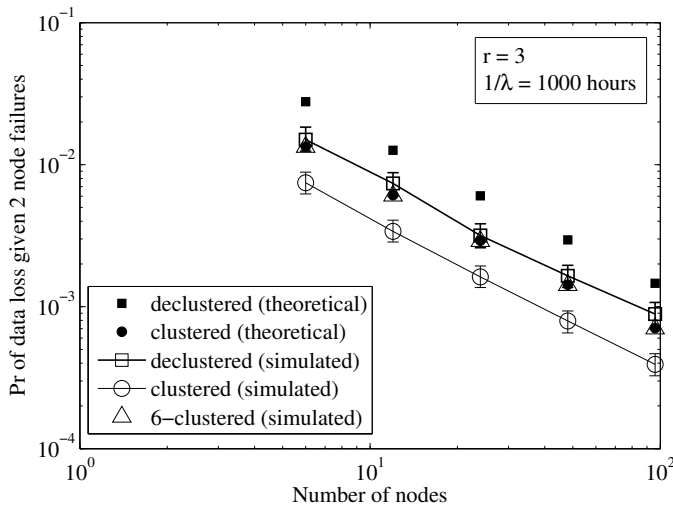
Fig. 4. Comparison of theoretically predicted and simulated values of probability of data loss given *two* node failures for a system with a replication factor of three and mean time to failure of a node equal to 1000 h; For the simulated results, 95% bootstrap confidence intervals are shown.
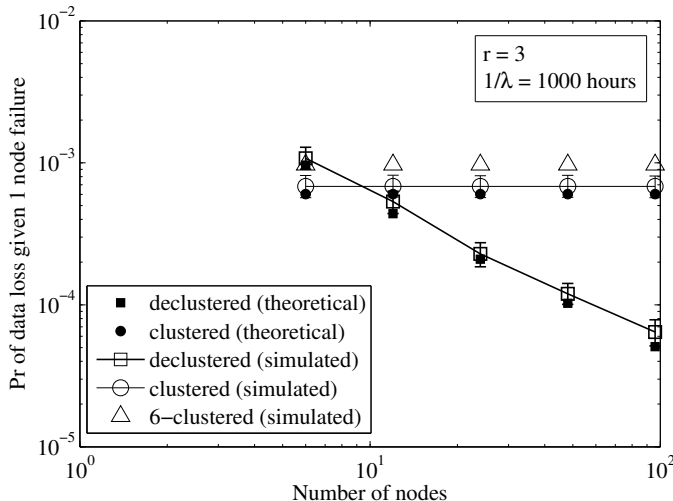

Fig. 5. Comparison of theoretically predicted and simulated values of probability of data loss given *one* node failure for a system with a replication factor of three and mean time to failure of a node equal to 1000 h; For the simulated results, 95% bootstrap confidence intervals are shown.

theoretical calculation of which is work in progress, appears to better predict the MTTDL for a replication factor of three. Note that, for a replication factor of two, this measure is the same $\mathscr{P}$.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we showed that all placement schemes have MTTDL values that differ by at most a factor of two for a practical storage system using a replication factor of two. We used a measure of reliability that is not only simple enough to enable sufficient analytical tractability but also comprehensive enough to take into account the effect of the placement schemes on the rebuild process. The measure used, namely, the probability of data loss during rebuild in critical mode, is shown to be affected by at most a factor of two by the choice of placement scheme for any replication factor. However, simulation results reveal that this property also holds for the corresponding MTTDLs, but only for a replication factor of two. For higher replication factors, the measure used is not a suitable indicator of the MTTDL behavior. This suggests that alternate measures of reliability are not always appropriate for comparing the MTTDL of different placement schemes.

We also show that the clustered placement scheme has the lowest probability of data loss during rebuild in critical mode, whereas the declustered placement scheme has the highest probability. This particular result is consistent with results of [4], [5], [6]. However, it differs from the results of [7], [8], [9]. We believe that the inconsistency with the latter set of publications is mainly because their analysis is based on an idealistic assumption that the replica sets (referred to as redundancy sets or groups in [7], [8], and as objects in [9]) fail independently. In contrast, in our analysis we take into account the correlations among the failures of different replica sets that are induced by node failures.

It is likely that the probability of data loss given the first node failure is a suitable measure that can be used to compare MTTDLs of different placement schemes. However, it is still to be seen whether this probability is as intractable as MTTDL or not. On the other hand, we conjecture that the core results of this paper extend beyond replication-based systems. It is likely that such results also exist for general erasure codes.

APPENDIX A
PROOF OF LEMMA VI.1

Let the total number of data blocks read from and written to each node be given by the distribution $\mathbf{v} := \mathbf{w} + \mathbf{w}'(\mathbf{w})$. Let the set of all $\mathbf{v}$ for a given number of critical blocks w be denoted by $\mathcal{V}(w)$, that is, $\mathcal{V}(w) := \{\mathbf{v}|w \text{ critical blocks}\}$.

Lemma A.1. *The set $\mathcal{V}(w)$ has the following properties:*
  (i) $0 \leq v_i \leq w, \, \forall \, i \in \{1, \cdots, n-r+1\}, \, \forall \, \mathbf{v} \in \mathcal{V}(w)$,
 (ii) $\sum_i v_i = 2w, \, \forall \, \mathbf{v} \in \mathcal{V}(w)$,
(iii) *The corner points of the convex hull of $\mathcal{V}(w)$ are $\{\mathbf{v}|v_j = v_k = w \text{ for some } j, k \text{ and } v_i = 0 \, \forall \, i \neq j, k\}$.*

Proof: (i) For any node i, $v_i \geq 0$ as it is the sum of the number of blocks, which is always non-negative. The total number of critical blocks is w and the new replicas of critical blocks are created such that no two replicas of the same block lie on the same node. This means that, the number $v_i = w_i + w_i'$ for any node i cannot be greater than w, because if the sum is greater than w, by the Pigeonhole principle, there has to be at least one block with two of its replicas on the same node.
(ii) $\sum_i v_i = \sum_i w_i + \sum_i w_i' = w + w = 2w$.
(iii) The corner points of the convex hull of $\mathcal{V}(w)$ are exactly the points where v_i's are allowed to take the extremal values of 0 and w. As the sum of all v_i should be $2w$ according to Property (ii), the corner points are given the by set of all $\mathbf{v}$, where $v_j = v_k = w$ for some $j, k \in \{1, \cdots, n-r+1\}$ and $v_i = 0 \, \forall \, i \neq j, k$. ∎

A. Upper Bound

To obtain the upper bound in Lemma VI.1, it suffices to find (see (17))

$$\max_{\mathbf{w}:\sum w_i = w} \max_{\mathbf{v}\in\mathcal{V}(w)} \prod_{i\in I(\mathbf{w})}\left(1 + \frac{\lambda s}{b}v_i\right).$$

Then, as all terms are inside the product above are non-negative by Lemma A.1 Property (i), we use the Arithmetic-Geometric Mean Inequality to get

$$\prod_{i\in I(\mathbf{w})}\left(1 + \frac{\lambda s}{b}v_i\right) \leq \left(1 + \frac{\lambda s}{b}\frac{\sum_i v_i}{|I(\mathbf{w})|}\right)^{|I(\mathbf{w})|}$$

$$= \left(1 + \frac{2\lambda s w}{b|I(\mathbf{w})|}\right)^{|I(\mathbf{w})|},$$

where the second step follows from Lemma A.1 Property (ii). Equality holds above when all v_i's are equal. As the sum of all v_i's is $2w$, equality holds when $v_i = w_i + w_i' = \frac{2w}{n-r+1}\,\forall i$. Plugging the above inequality into (17), we get

$$\Pr\{\mathrm{DL}|\mathbf{w},\mathbf{w}'(\mathbf{w})\}$$

$$\leq 1 - \min_{\mathbf{w}:\sum w_i = w}\left(1 + \frac{2\lambda s w}{b|I(\mathbf{w})|}\right)^{-|I(\mathbf{w})|}.$$

By Taylor's theorem, it can be shown that

$$\left(1 + \frac{2\lambda s w}{b|I(\mathbf{w})|}\right)^{-|I(\mathbf{w})|} > 1 - \frac{2\lambda s w}{b}. \tag{21}$$

Therefore, $\Pr\{\mathrm{DL}|\mathbf{w},\mathbf{w}'(\mathbf{w})\} < \frac{2\lambda s w}{b}$.

B. Lower Bound

To obtain the lower bound in Lemma VI.1, it suffices to find (see (17))

$$\min_{\mathbf{w}:\sum w_i = w} \min_{\mathbf{v}\in\mathcal{V}(w)} \underbrace{\prod_{i\in I(\mathbf{w})}\left(1 + \frac{\lambda s}{b}v_i\right)}_{=:\,f(\mathbf{v})}.$$

The function $f(\mathbf{v})$ is a concave function defined on the convex hull of $\mathcal{V}(w)$. Therefore, the minimum of the function is attained at the corner points of the convex hull, which by Lemma A.1 Property (iii), are given by $\{\mathbf{v}\in\mathcal{V}(w)|v_j = v_k = w$ for some $j,k\in\{1,\cdots,n-r+1\}$ and $v_i = 0\,\forall\,i\neq j,k\}$. The above minimization problem thus reduces to finding

$$\min_{\mathbf{w}:\sum w_i = w} \prod_{i\in\{j,k\}\cap I(\mathbf{w})}\left(1 + \frac{\lambda s w}{b}\right).$$

The minimum is attained when $|\{j,k\}\cap I(\mathbf{w})| = 1$. Without loss of generality, let $\{j,k\}\cap I(\mathbf{w}) = \{j\}$. This implies $w_j = w$. Therefore, plugging the above minimum into (17), we get

$$\Pr\{\mathrm{DL}|\mathbf{w},\mathbf{w}'(\mathbf{w})\} \geq \frac{\lambda s w}{b}\frac{1}{\left(1 + \frac{\lambda s w}{b}\right)}, \tag{22}$$

where equality holds for all $\{\mathbf{w},\mathbf{w}'(\mathbf{w})|w_j = w$ for some j and $w_i = 0\,\forall\,i\neq j\}$.

REFERENCES

[1] C. Miller, A. R. Butt, and P. Butler, "On utilization of contributory storage in desktop grids," in *Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS'08)*, April 2008, pp. 1–12.

[2] K. Greenan, E. L. Miller, and J. Wylie, "Reliability of XOR-based erasure codes on heterogeneous devices," in *Proc. 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'08)*, June 2008, pp. 147–156.

[3] S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, 2006, pp. 1–9.

[4] M. Leslie, J. Davies, and T. Huffman, "A comparison of replication strategies for reliable decentralised storage," *Journal of Networks*, vol. 1, no. 6, pp. 36–44, December 2006.

[5] A. Thomasian and M. Blaum, "Mirrored disk organization reliability analysis," *IEEE Transactions on Computers*, vol. 55, pp. 1640–1644, December 2006.

[6] M. Jiang, J. Zhou, M. Hu, and Y. X. Ding, "Fuzzy reliability of mirrored disk organizations," in *Proc. 2007 International Conference on Convergence Information Technology (ICCIT'07)*, 2007, pp. 1345–1348.

[7] Q. Xin, E. L. Miller, T. Schwarz, D. D. E. Long, S. A. Brandt, and W. Litwin, "Reliability mechanisms for very large storage systems," in *Proc. 20th IEEE / 11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03)*, 2003, pp. 146–156.

[8] Q. Xin, E. L. Miller, and T. J. E. Schwarz, "Evaluation of distributed recovery in large-scale storage systems," in *Proc. 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04)*, 2004, pp. 172–181.

[9] Q. Lian, W. Chen, and Z. Zhang, "On the impact of replica placement to the reliability of distributed brick storage systems," in *Proc. 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005, pp. 187–196.

[10] IBM, "XiV Storage System Specifications." [Online]. Available: www.xivstorage.com

[11] W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky, "Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics," *ACM Transactions on Storage*, vol. 4, no. 3, pp. 1–25, November 2008.

[12] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, "Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems," in *Proc. 2008 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'08)*, 2008, pp. 241–252.

[13] B. Schroeder and G. A. Gibson, "Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you?" *ACM Transactions on Storage*, vol. 3, no. 3, pp. 1–31, October 2007.

[14] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *Proc. 5th USENIX conference on File and Storage Technologies (FAST'07)*, 2007, pp. 17–28.

[15] S. Nath, H. Yu, P. B. Gibbons, and S. Seshan, "Subtleties in tolerating correlated failures in wide-area storage systems," in *Proc. 3rd conference on Networked Systems Design & Implementation (NSDI'06)*, 2006, pp. 225–238.

[16] 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 Transactions on Storage*, vol. 4, no. 1, pp. 1–42, May 2008.

[17] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: A fast array of wimpy nodes," in *Proc. 22nd ACM Symposium on Operating Systems Principles (SOSP'09)*, October 2009.

[18] M. W. Storer, K. Greenan, E. L. Miller, and K. Voruganti, "Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage," in *Proc. 6th USENIX Conference on File and Storage Technologies (FAST'08)*, February 2008, pp. 1–16.