RZ 3821 (# Z1204-003) 04/17/2012
Computer Science 9 pages

Research Report

Reliability of Data Storage Systems under Network Rebuild Bandwidth Constraints

Vinodh Venkatesan, Ilias Iliadis, Robert Haas

IBM Research – Zurich
8803 Rüschlikon
Switzerland

Reliability of Data Storage Systems under Network Rebuild Bandwidth Constraints

Vinodh Venkatesan, Ilias Iliadis, and Robert Haas
IBM Research - Zurich
8803 Rüschlikon, Switzerland
{ven, ili, rha}@zurich.ibm.com

***Abstract*—To improve the reliability of data storage systems, certain data placement schemes spread replicas across several nodes. This enables parallelizing the rebuild process which in turn results in reducing the rebuild times. However, the underlying assumption is that the parallel rebuild process is facilitated by sufficient availability of network bandwidth to transfer data across nodes. In a large-scale data storage system where the network bandwidth for rebuild is constrained, such placement schemes will not be as effective. In this paper, it is shown through analysis and simulation how the spread of replicas across nodes affects system reliability under a system network bandwidth constraint. Efficient placement schemes that can achieve high reliability in the presence of bandwidth constraints are proposed. Furthermore, in a dynamically changing storage system, in which the number of nodes and the network rebuild bandwidth can change over time, the data placement can be accordingly adapted to maintain the highest level of reliability.**

I. Introduction

Redundancy is used to protect data from node failures in today's large-scale data storage systems. When a storage node fails, the lost redundancy is restored as fast as possible through a rebuild process. This restoration is done by reading the redundancies corresponding to the lost data from other surviving nodes, reconstructing the lost data (which could either involve some computation if complex erasure codes are used for redundancy, or simple copying if replication is used), and storing the reconstructed data either in the spare space of surviving nodes or in spare nodes. The reliability of a system, in terms of the chances of this rebuild process failing due to further node failures resulting in irrecoverable data loss, depends significantly on this rebuild process.

The manner in which the redundant data are placed across the nodes in the system, that is, data placement, affects both how fast and how effective the rebuild process can be. There are two main ways in which the data placement, and hence the rebuild process, affects the reliability of the system. Firstly, if the redundant data are placed across several nodes in the system, the rebuild process can benefit by parallelizing the data restoration process. The restoration time can be minimal provided there is sufficient network bandwidth available. Minimal restoration time implies that there is a shorter window of time during which additional node failures can hinder the rebuild. Secondly, spreading the replicas of data across several nodes also exposes these replicas to the failure of any of these nodes, thereby increasing the probability of failure of the rebuild process. Interestingly, for two-way replicated systems, these two effects cancel each other out resulting in similar reliability [1]. For higher replication factors, however, the first effect is more dominant as the second effect tends to expose less amount of data to the danger of irrecoverable loss because of the spreading of replicas across several nodes [2]. When the network bandwidth is limited though, the rebuild times may be longer and therefore the former factor may be affected. This imbalance leads to interesting results in terms of the mean time to data loss (MTTDL) of the system. In this paper, we explore this effect and show how the network bandwidth constraint affects the system MTTDL and how we can design schemes that can achieve high reliability under these conditions. The results of this paper can also be used to adapt the data placement schemes when the available network rebuild bandwidth or the number of nodes in the system changes so that system reliability is maintained at the highest level.

The remainder of this paper is organized as follows. Section II briefly reviews related work in the literature. Section III lists the system parameters, and describes the node failure and rebuild models used. Section IV presents the method of reliability estimation. Section V contains the main results of this paper on the effect of network rebuild bandwidth and the data placement scheme used on the system reliability. Section VI lists the findings based on simulations and theory, and Section VII concludes this paper.

II. Related Work

Dependence of system reliability on the data placement scheme without any network rebuild bandwidth limitations has been studied extensively in the literature [1], [2], [3], [4], [5], [6]. Effect of network rebuild bandwidth constraints on system reliability has been studied in [5]. That work considers and assesses the reliability of two placement schemes, namely, sequential and random. In sequential placement, for any data block there is one unique node that acts as a lead node, and the r replicas of this data block are stored on the lead node and its $r-1$ following nodes. This implies that the replicas corresponding to the data on any given storage node are spread over its $r-1$ preceding nodes and its $r-1$ following nodes. This is equivalent to a placement scheme with replica spread factor equal to $2r+1$ (see Section III-B). In random placement, the r replicas of a data block are placed randomly across n nodes. If the size of the data block is small enough,

TABLE I
PARAMETERS OF A STORAGE SYSTEM

c	storage capacity of each node (bytes)
n	number of storage nodes
$1/\lambda$	mean time to failure of a storage node (s)
b	rebuild bandwidth at each storage node (bytes/s)
r	replication factor
k	spread factor of the data placement scheme
B_{max}	maximum network rebuild bandwidth (bytes/s)
$1/\mu$	time to rebuild a node in clustered placement ($1/\mu = c/b$)
N	maximum number of nodes from which rebuild can occur at full speed in parallel ($N = B_{\text{max}}/b$)
$B_{\text{eff}}(\tilde{k})$	effective distributed rebuild bandwidth involving $\tilde{k}$ nodes ($B_{\text{eff}}(\tilde{k}) = \min(\tilde{k}b, B)$)
$S_{\text{eff}}(\tilde{k})$	effective speed of distributed rebuild involving $\tilde{k}$ nodes ($S_{\text{eff}}(\tilde{k}) = B_{\text{eff}}(\tilde{k})/2$)

then all possible placement choices of r out of n nodes will be used and this is equivalent to declustered placement with replica spread factor equal to n (see Section III-B). However, if the size of data blocks is large, then all possible placement choices of r out of n nodes may not be used by the random placement scheme and the spread factor may be smaller than n. In addition, [5] proposes a new placement scheme called *stripe placement* which essentially limits the spread factor to the maximum number of nodes that the network rebuild bandwidth can support during a parallel rebuild process at full speed. Our results closely match the mean time to data loss (MTTDL) results of [5]. In addition, we derive closed form expressions for the MTTDL and provide further insight into the effect of placement schemes on the reliability behavior of systems under network rebuild bandwidth constraints.

III. SYSTEM MODEL

The parameters of the storage system considered and the failure and rebuild models used in the paper are described in this section. Table I lists the parameters used. The upper and lower parts of the table list the set of independent and dependent parameters.

A. Storage System

Consider a storage system with n nodes each with capacity c. Data redundancy across nodes is used to protect data from node failures in the system. In this paper, we consider a simple form of redundancy, namely, replication, although many of the results in this paper can be extended to other forms of redundancy such as erasure codes as well. When a node failure occurs, a rebuild process is initiated to restore the lost data from the surviving replicas. Data loss occurs when a series of node failures occur that eventually makes some data lose all its r replicas.

B. Data Placement Schemes

For each node in the system, let its *redundancy spread factor* denote the number of nodes over which the data on that node and its corresponding redundant data are spread. In a replication-based system, when a node fails, its spread factor determines the number of nodes which have replicas of the data in the failed node, and this in turn determines the degree of parallelism that can be used in rebuilding the data lost by that node. In this paper, we will consider symmetric placement schemes in which the spread factor of each node is the same, denoted by k. 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. One example of such a symmetric placement scheme is the so-called *clustered placement* scheme for which the spread factor k is equal to r. In this scheme, the system is divided into disjoint sets of r nodes, called clusters, and the nodes in each cluster store replicas of the same data. Another example of a symmetric placement scheme is the so-called *declustered placement* scheme for which the spread factor k is equal to n. In this scheme, all $\binom{n}{r}$ possible ways of placing r replicas in n nodes are equally used to store data. A number of different placement schemes can be generated by varying the spread factor k. The spread factor of a placement scheme is important in two ways: (a) it determines the number of nodes over which data of a failed node is spread and therefore, the degree of parallelism that can be used in the rebuild process of that node, and (b) it determines the amount of data that becomes critical and needs to be rebuilt first when additional node failures occur. It can be seen that any two nodes sharing replicas of some data share exactly $\frac{r-1}{k-1}c$ amount of data. In general, any set of m nodes ($m < r$) sharing replicas of some data, share exactly $c\prod_{i=1}^{m-1}\left(\frac{r-i}{k-i}\right)$ amount of data.

C. Failure Model

Times to node failures are assumed to be independent and exponentially distributed with mean time to failure $1/\lambda$. It can be shown that the MTTDL estimates are practically insensitive to a large class of failure distributions, including most importantly, real world distributions [7]. Therefore, the MTTDL results derived in this paper using exponential distributions also apply to real world failure time distributions. The independence assumption 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 [8]. Recent work [9] has shown that node *unavailability* can be strongly correlated; however, there is no specific characterization of the extent of correlation among *permanently failing* nodes.

D. Rebuild Model

The rebuild process used to restore the data lost by failed nodes is assumed to be both *intelligent* and *distributed*. By an intelligent rebuild process, we mean that the system always attempts to first recover the copies (replicas) of the most critical data, that is, data that has the least number, say $\tilde{r}$, of replicas left in the system. In a distributed rebuild process, the data lost by a failed node is restored by reading surviving replicas and creating a new replica of the data in reserved spare space on surviving nodes as illustrated in Fig. 1. More specifically, if the $\tilde{r}$ surviving replicas of the most critical data

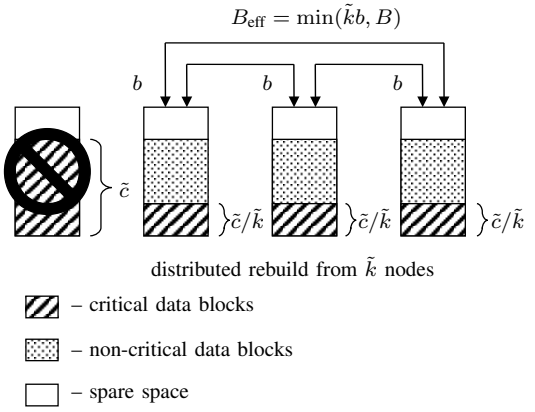

Fig. 1. Example of the distributed rebuild model for a two-way replicated system. When one node fails, the critical data blocks are present in the surviving nodes. The distributed rebuild process creates replicas of these critical blocks by copying them from one surviving node to another in parallel.

are stored across $\tilde{k}$ nodes ($\tilde{k} > \tilde{r}$), these replicas are used to rebuild the lost data in the spare space on those $\tilde{k}$ nodes such that no two copies of the same data are stored on the same node. This is done so that the rebuild process can make use of the node rebuild bandwidth available at all $\tilde{k}$ nodes in parallel. Once all lost data is recovered, this newly recovered data is transferred to new nodes. For clustered placement, the surviving $\tilde{r}$ replicas of the most critical data of a cluster are present on exactly the $\tilde{r}$ surviving nodes of that cluster, that is, $\tilde{k} = \tilde{r}$. Therefore, the replicas of this data are read from one of the surviving nodes and written to a new spare node as it is not possible to do a distributed rebuild as described earlier without creating two replicas of the same data on the same node.

During the rebuild process, a read-write bandwidth of b bytes/s is assumed to be reserved at each node exclusively for the rebuild. This is usually only a fraction of the total bandwidth available at each node; the remainder is being used to serve user requests. If $1/\mu$ is the time to rebuild a storage node in clustered placement, that is, the time required to read all contents of a node and write to a new spare node, then $1/\mu = c/b$ or $b = c\mu$.

In a distributed rebuild process, if the $\tilde{r}$ surviving replicas of the most critical data are stored across $\tilde{k}$ nodes, then the total network bandwidth required to perform rebuild at full speed is $\tilde{k}b$. Let the maximum available network bandwidth for rebuilds be denoted by B_{max}. We will assume that $B_{\text{max}} \geq b$ as $B_{\text{max}} < b$ is a degenerate case. So, if the available network rebuild bandwidth is B_{max}, the total bandwidth that can be used by rebuilds cannot exceed B_{max}. Therefore, the effective network rebuild bandwidth used by rebuilds, $B_{\text{eff}}(\tilde{k})$, is given by

$$B_{\text{eff}}(\tilde{k}) = \min(\tilde{k}b, B_{\text{max}}) = \min(\tilde{k}, N)b, \tag{1}$$

where N specifies the effective maximum number of nodes from which rebuild can occur in parallel at full speed and is given by

$$N = \frac{B_{\text{max}}}{b}. \tag{2}$$

Note that N may not be an integer; it only represents the *effective* maximum number of nodes from which distributed rebuild can occur at full speed. Substituting $b = c\mu$ into (1), we get

$$B_{\text{eff}} = \min(\tilde{k}, N)c\mu. \tag{3}$$

Suppose that the total amount of critical data to be rebuilt is $\tilde{c}$. Owing to the nature of the symmetric placement and the nature of the distributed rebuild process, the amounts of data to be read from and to be written to each of the $\tilde{k}$ nodes are equal to $\tilde{c}/\tilde{k}$. As the effective rebuild bandwidth $B_{\text{eff}}(\tilde{k})$ is equally used for both reads and writes, the required rebuild time, $\tilde{R}$, is given by

$$\tilde{R} = \tilde{c}/(B_{\text{eff}}(\tilde{k})/2) = \tilde{c}/S_{\text{eff}}(\tilde{k}), \tag{4}$$

where $S_{\text{eff}}(\tilde{k})$ is the effective speed (rate) of distributed rebuild involving $\tilde{k}$ nodes and is given by

$$S_{\text{eff}}(\tilde{k}) = B_{\text{eff}}(\tilde{k})/2. \tag{5}$$

For the sake of clarity and consistency with earlier works, we will only use expressions involving μ and N rather than b and B_{max} in the remainder of the paper. The implicit relationship between μ, N, b, and B_{max} is given in Table I.

Clustered placement is an exception as it does not use distributed rebuild. The effective speed of rebuild for clustered placement is $c\mu$ because data is read from any *one* of the surviving nodes of the cluster to which the failed node belonged, and then written to a spare node.

In this paper, as in [1], [2], we will assume that the rebuild bandwidth is constant and hence the rebuild times are fixed. However, the results obtained here can be extended to a large class of rebuild time distributions by using the methodology presented in [7].

E. Generally Reliable Nodes

We will assume that storage nodes are *generally reliable*, that is, the mean time to failure of a node $1/\lambda$ is much larger than the time to rebuild a node $1/\mu$:

$$1/\lambda \gg 1/\mu, \quad \text{or} \quad \lambda/\mu \ll 1. \tag{6}$$

This condition implies that terms involving powers of λ/μ greater than one are negligible compared to λ/μ and can be ignored in the subsequent analysis.

IV. Reliability Estimation

At any point in time, the system is in one of two modes, namely, fully-operational mode or rebuild mode. In the fully-operational mode, all data in the system have r replicas and there are no rebuilds in progress. In the rebuild mode, some data in the system have lost some of their replicas and a rebuild process that attempts to restore the lost replicas is underway. A transition from fully-operational mode to rebuild mode occurs when a node fails; we refer to this node failure that causes this

transition as a *first-node failure*. Following a first-node failure, a complex sequence of rebuilds and subsequent node failures may occur which eventually lead the system either back to the original fully-operational mode or to irrecoverable data loss. As the storage nodes are generally reliable, the rebuild times are much shorter than the times to failure, and therefore the time taken for this complex sequence of events is negligible compared to the time to first-node-failure. Consequently, we model the systems behavior as a series of first-node-failure events each of which results in data loss with probability P_{DL}, or back to the original fully-operational mode with probability $1 - P_{DL}$.

As the times to failure of the nodes are exponentially distributed with mean $1/\lambda$, the mean time between two first-node failures is equal to $1/(n\lambda)$. Furthermore, as each first-node failure could result in data loss with probability P_{DL}, the expected number of first-node failures until data loss occurs is $1/P_{DL}$. Therefore, the mean time to data loss (MTTDL) is equal to the product of the mean time between two first-node failures and the expected number of first-node failures, that is,

$$\text{MTTDL} \approx \frac{1}{n\lambda P_{DL}}. \tag{7}$$

The above expression is approximate because the rebuild times, which are negligible compared to the time between failures, are ignored. Note that the above expression also holds for more general non-exponential failure distributions [7].

It has been argued that MTTDL is useful for assessing trade-offs, for comparing schemes, and for estimating the effect of the various parameters on the system reliability [10], [11], [12]. Since the main objective of this work is the assessment of the effect of the network rebuild bandwidth on the reliability of various types of data placement schemes, we proceed by considering MTTDL as a reliability measure.

V. Effect of Network Rebuild Bandwidth

As noted earlier, limited network rebuild bandwidth can negatively influence the rebuild process and therefore lower the reliability of the system. In this section, we analyze its effect by estimating the MTTDL of a system with spread factor $k > r$. The analysis is done by modeling the system using *exposure levels* and computing the probability P_{DL} [2].

A. Exposure Levels

At time t, let $D_l(t)$ be the number of distinct data blocks that have lost l replicas, with $0 \leq l \leq r$. The system is considered to be in exposure level e at time t, $0 \leq e \leq r$, if

$$e = \max_{D_l(t)>0} l. \tag{8}$$

In other words, the system is in exposure level e if there exists at least one block with $r-e$ copies and no blocks 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 number of distinct data blocks stored in the system. Node failures and rebuild processes cause the values of $D_1(t), \cdots, D_r(t)$ to change over time, and when data loss occurs, $D_r(t) > 0$.

B. Direct Path to Data Loss

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

$$P_{DL} \approx P_{DL,\text{direct}} = \prod_{e=1}^{r-1} P_{e\to e+1}, \tag{9}$$

where $P_{e\to e+1}$ denotes the probability of transition from exposure level e to $e+1$.

C. Rebuild Times at Each Exposure Level

Consider the direct path to data loss and let the rebuild times of the most-exposed data at each exposure level in this path be denoted by R_e, $e = 1, \cdots, r-1$. Let α_e be the fraction of the rebuild time R_e still left when a node failure occurs causing an exposure level transition. It has been shown in [2] that α_e is uniformly distributed, that is,

$$\alpha_e \sim U(0,1), \quad e = 1, \cdots, r-2. \tag{10}$$

The amount of data to be rebuilt in exposure level 1 is c. For a placement scheme with spread factor $k > r$, the surviving replicas of this data are spread across $k-1$ nodes. Therefore, the speed of distributed rebuild follows from (5) and is equal $S_{\text{eff}}(k-1)$. The time to rebuild follows from (4) and is given by

$$R_1 = \frac{c}{S_{\text{eff}}(k-1)}. \tag{11}$$

Given α_1 and R_1, the remaining time to complete the rebuild is $\alpha_1 R_1$. As the rate of rebuild is $S_{\text{eff}}(k-1)$, it now follows that the amount of data *not* rebuilt in exposure level 1 is $\alpha_1 R_1 S_{\text{eff}}(k-1)$. However, copies of only a fraction, $\frac{r-1}{k-1}$, of this data were shared by the newly failed node due to the nature of the symmetric placement scheme. This implies that, in exposure level 2, the amount of most-exposed data, that is, the data which have lost 2 copies, is $\frac{r-1}{k-1}\alpha_1 R_1 S_{\text{eff}}(k-1)$. The system performs intelligent rebuild, that is, rebuilds the most-exposed data first. By the nature of the placement scheme, the surviving replicas of the most-exposed data are now spread across $k-2$ nodes and therefore the speed of distributed rebuild follows from (5) and is equal to $S_{\text{eff}}(k-2)$. The time to rebuild in exposure level 2, R_2, follows from (4) and is given by

$$\begin{aligned} R_2 &= \frac{\frac{r-1}{k-1}\alpha_1 R_1 S_{\text{eff}}(k-1)}{S_{\text{eff}}(k-2)} \\ &= \frac{r-1}{k-1} \cdot \frac{S_{\text{eff}}(k-1)}{S_{\text{eff}}(k-2)} \alpha_1 R_1. \end{aligned} \tag{12}$$

Using similar arguments, for any given exposure level e, $e = 2, \cdots, r-1$, it holds that

$$R_e = \frac{r-e+1}{k-e+1} \cdot \frac{S_{\text{eff}}(k-e+1)}{S_{\text{eff}}(k-e)} \alpha_{e-1} R_{e-1}. \tag{13}$$

D. Conditional Probability of Exposure Level Transition

In the direct path to data loss, there are $k-e$ nodes in exposure level e whose failure before rebuild can cause the system to go to exposure level $e+1$. Denote the times to failure of these nodes by $E_t^{(i)}$, $i=1,\cdots,k-e$, and denote by $P_{e\to e+1}(R_e)$ the conditional probability of transition to exposure level $e+1$ given that the rebuild time is R_e. Then,

$$\begin{aligned} P_{e\to e+1}(R_e) &= \Pr\{\min_i E_t^{(i)} \le R_e\} \\ &= 1-(1-\Pr\{E_t^{(1)} \le R_e\})^{k-e}. \quad (14) \end{aligned}$$

As $E_t^{(i)}$, $i=1,\cdots,k-e$ are exponentially distributed with mean $1/\lambda$,

$$\Pr\{E_t^{(1)} \le R_e\} \approx \lambda R_e. \quad (15)$$

The above approximation for the value of $\Pr\{E_t^{(1)} \le R_e\}$ also holds for more general non-exponential distribution [7]. Substituting (15) into (14), and ignoring higher powers of λR_e, we get

$$P_{e\to e+1}(R_e) \approx (k-e)\lambda R_e. \quad (16)$$

E. Estimation of MTTDL

Consider a realization of the direct path to data loss with fractions α_e, $e=1,\cdots,r-2$, and rebuild times R_e, $e=1,\cdots,r-1$. For notational convenience, let us denote the vector $(\alpha_1,\cdots,\alpha_{r-2})$ by $\vec{\alpha}$, $(R_1,\cdots,R_{r-1})$ by $\vec{R}$, and the conditional probability of this path by $P_{DL,\text{direct}}(\vec{\alpha},\vec{R})$. Then, using (16),

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\alpha},\vec{R}) &= \prod_{e=1}^{r-1} P_{e\to e+1}(R_e) \\ &\approx \lambda^{r-1}\prod_{e=1}^{r-1}(k-e)R_e. \quad (17) \end{aligned}$$

By unconditioning on $\vec{\alpha}$ and $\vec{R}$, we now obtain

$$\begin{aligned} P_{DL,\text{direct}} &= E\left[P_{DL,\text{direct}}(\vec{\alpha},\vec{R})\right] \\ &\approx \lambda^{r-1}E[R_1R_2\cdots R_{r-1}]\prod_{e=1}^{r-1}(k-e). \quad (18) \end{aligned}$$

Then by the direct path approximation (9), the probability P_{DL} is given by

$$P_{DL} \approx \lambda^{r-1}E[R_1R_2\cdots R_{r-1}]\prod_{e=1}^{r-1}(k-e). \quad (19)$$

Substituting (13) for $e=2,\cdots,r-1$ in (19), we obtain

$$\begin{aligned} P_{DL} \approx (\lambda R_1)^{r-1}E[\alpha_1^{r-2}\alpha_2^{r-3}\cdots\alpha_{r-3}^2\alpha_{r-2}]\prod_{e=1}^{r-1}(k-e) \\ \times\prod_{e'=2}^{r-1}\left(\frac{r-e'+1}{k-e'+1}\frac{S_{\text{eff}}(k-e'+1)}{S_{\text{eff}}(k-e')}\right)^{r-e'}. \quad (20) \end{aligned}$$

Substituting (11) into (20), and using the fact that $\alpha_e \sim U(0,1)$, $e=1,\cdots,r-2$ and therefore $E[\alpha_e^m]=1/(m+1)$, we get

$$\begin{aligned} P_{DL} \approx \left(\frac{\lambda c}{S_{\text{eff}}(k-1)}\right)^{r-1}\frac{1}{(r-1)!}\prod_{e=1}^{r-1}(k-e) \\ \times\prod_{e'=2}^{r-1}\left(\frac{r-e'+1}{k-e'+1}\frac{S_{\text{eff}}(k-e'+1)}{S_{\text{eff}}(k-e')}\right)^{r-e'}. \quad (21) \end{aligned}$$

Substituting (5) and (3) into (21), we get

$$\begin{aligned} P_{DL} \approx \left(\frac{\lambda}{\mu}\frac{2}{\min(k-1,N)}\right)^{r-1}\frac{1}{(r-1)!}\prod_{e=1}^{r-1}(k-e) \\ \times\prod_{e'=2}^{r-1}\left(\frac{r-e'+1}{k-e'+1}\frac{\min(k-e'+1,N)}{\min(k-e',N)}\right)^{r-e'}. \quad (22) \end{aligned}$$

Canceling terms of the form $\min(k-e',N)$ and rewriting the product, we get

$$\begin{aligned} P_{DL} \approx \left(\frac{2\lambda}{\mu}\right)^{r-1}\frac{1}{(r-1)!}\prod_{e=1}^{r-2}\left(\frac{r-e}{k-e}\right)^{r-e-1} \\ \times\prod_{e'=1}^{r-1}\frac{k-e'}{\min(k-e',N)}. \quad (23) \end{aligned}$$

An estimate for the MTTDL then follows by substituting (23) into (7):

$$\begin{aligned} \text{MTTDL} &\approx \frac{\mu^{r-1}}{n\lambda^r}\frac{(r-1)!}{2^{r-1}}\prod_{e=1}^{r-2}\left(\frac{k-e}{r-e}\right)^{r-e-1} \\ &\quad\times\prod_{e'=1}^{r-1}\frac{\min(k-e',N)}{k-e'}. \quad (24) \end{aligned}$$

The above expression holds for all spread factors $k>r$. For $k=r$, that is, for clustered placement, the rebuild process always involves reading data from one of the surviving nodes and writing to a new node at a rate $c\mu$. Given that $B_{\max} > b = c\mu$, the network rebuild bandwidth $B_{\max}$ is sufficient to read and transfer data from one node to another at a rate $c\mu$, the MTTDL is given by [2]

$$\text{MTTDL}^{\text{clus.}} = \frac{\mu^{r-1}}{n\lambda^r}. \quad (25)$$

F. MTTDL vs. Network Rebuild Bandwidth

The expression for MTTDL in (24) can be broken down as follows to understand the effect of limited network rebuild bandwidth. When the spread factor $k \le N+1$, the terms of the form $\min(k-e,N)$ become equal to $k-e$ for $e=1,\cdots,r-1$, and the second product in expression (24) for MTTDL becomes equal to one. This represents the case when network rebuild bandwidth is sufficient and does not affect the reliability of the system. For $k=n$, that is, for declustered placement, and for $k \le N+1$, expression (24) is the same as expression (20) in [2]. On the other hand, when the spread factor $k \ge N+r-1$, the network rebuild bandwidth is insufficient for a parallel rebuild process at full speed and

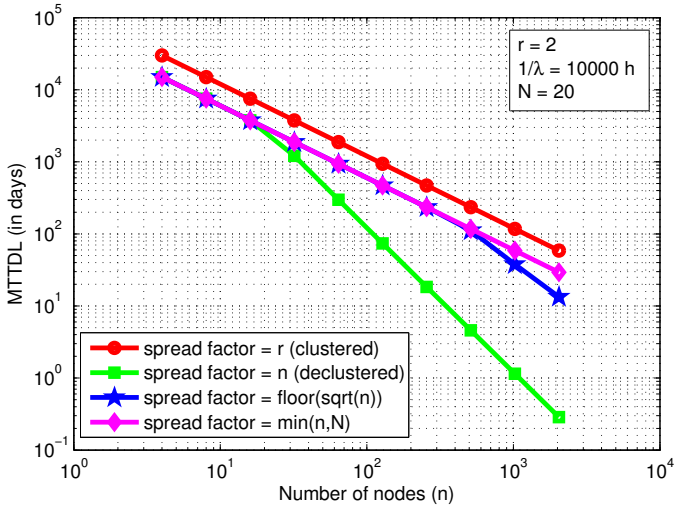

Fig. 2. Theoretical estimates of MTTDL for a replication factor of two with mean time to failure of a node equal to 10000 h.

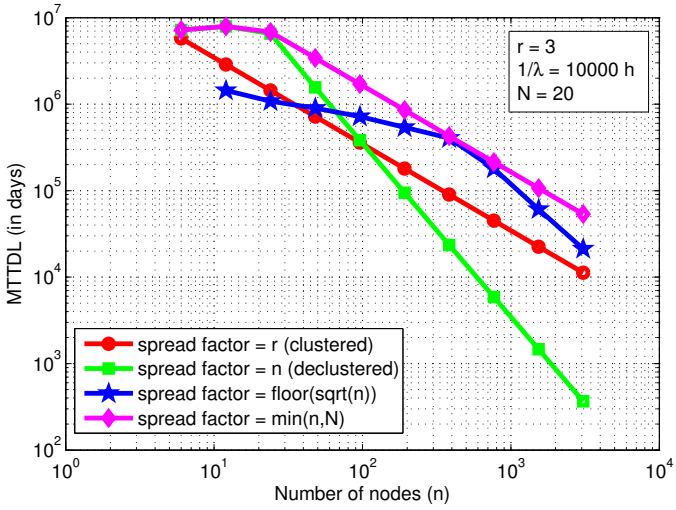

Fig. 3. Theoretical estimates of MTTDL for a replication factor of three with mean time to failure of a node equal to 10000 h.

therefore the system reliability is affected negatively. This can be seen from the fact that the second product in expression (24) for MTTDL becomes smaller than one and scales as $k^{-(r-1)}$.

In essence, if we denote the MTTDL under network bandwidth constraint N by MTTDL_N and the MTTDL under no network bandwidth constraint, that is, $N = \infty$, by MTTDL_∞, then it follows from (24) that

$$\text{MTTDL}_N = \text{MTTDL}_\infty \prod_{e'=1}^{r-1} \frac{\min(k-e', N)}{k-e'}, \tag{26}$$

where MTTDL_∞ is given by

$$\text{MTTDL}_\infty \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{(r-1)!}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{k-e}{r-e}\right)^{r-e-1}. \tag{27}$$

Replication Factor 2: For declustered placement, that is, for $k = n$, the expression for MTTDL (24) reduces to

$$\text{MTTDL}^{\text{declus.}} = \begin{cases} \dfrac{\mu}{2n\lambda^2} & \text{when } n \le N+1 \\ \dfrac{\mu N}{2n(n-1)\lambda^2} & \text{when } n \ge N+1. \end{cases} \tag{28}$$

The above expressions show that, when network rebuild bandwidth is not sufficient to carry out the rebuild process in parallel at full speed, the MTTDL becomes inversely proportional to the square of the number of nodes instead of being inversely proportional to the number of nodes. This drastic change in the MTTDL behavior as the system scales is shown in Fig. 2. The figure shows the plots of MTTDL as a function of the number of nodes for four different placement schemes, each with a different spread factor. When the network rebuild bandwidth can support only up to $N = 20$ nodes at full speed during distributed rebuild, it is seen that the MTTDL of declustered placement drops significantly compared to other placement schemes which are not affected (because their spread factors are less than N). For a scheme whose spread factor is $k = \lfloor \sqrt{n} \rfloor$, the change in MTTDL behavior is seen around $n = N^2 = 400$ nodes.

Replication Factor 3: For declustered placement, the expression for MTTDL (24) reduces to

$$\text{MTTDL}^{\text{declus.}} = \begin{cases} \dfrac{\mu^2(n-1)}{4n\lambda^3} & \text{when } n \le N+1 \\ \dfrac{\mu^2 N^2}{4n(n-2)\lambda^3} & \text{when } n \ge N+2. \end{cases} \tag{29}$$

The change in the MTTDL behavior due to limited network rebuild bandwidth is greater than that observed for replication factor two; it goes from being constant with respect to the number of nodes when network rebuild bandwidth is sufficient, to being inversely proportional to the square of the number of nodes when the network rebuild bandwidth is limited. This is also shown in Fig. 3. Interestingly, for $r = 3$, limiting the spread factor to N, that is, setting $k = \min(n, N)$, can achieve much higher MTTDL than the declustered placement scheme for $n \ge N + 2$.

Replication Factor 4: For declustered placement, the expression for MTTDL (24) reduces to

$$\text{MTTDL}^{\text{declus.}} = \begin{cases} \dfrac{\mu^3(n-1)^2(n-2)}{24n\lambda^4} & \text{when } n \le N+1 \\ \dfrac{\mu^3 N^2(N+1)}{24(N+2)\lambda^4} & \text{when } n = N+2 \\ \dfrac{\mu^3(n-1)N^3}{24n(n-3)\lambda^4} & \text{when } n \ge N+3. \end{cases} \tag{30}$$

The above expressions are plotted in Fig. 4. For replication factor 4, the MTTDL values of a scheme that limits the spread factor to N, that is, $k = \min(n, N)$, is comparable to the MTTDL values of the declustered scheme for which $k = n$. This is because, although the limited network bandwidth slows down rebuilds in a declustered placement scheme, the

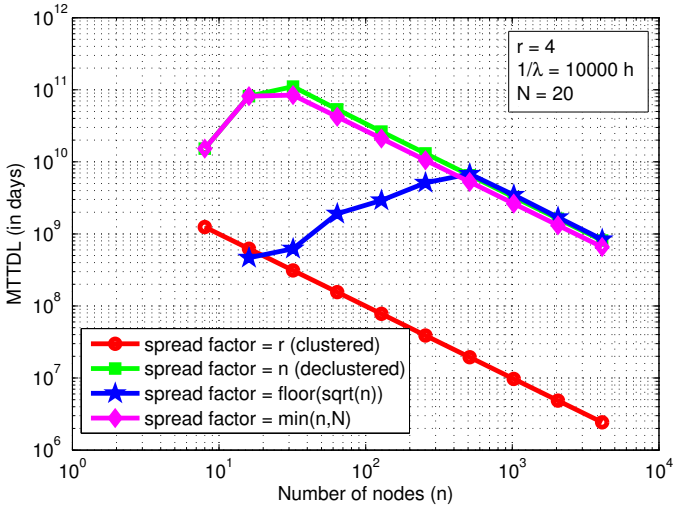

Fig. 4. Theoretical estimates of MTTDL for a replication factor of four with mean time to failure of a node equal to 10000 h.

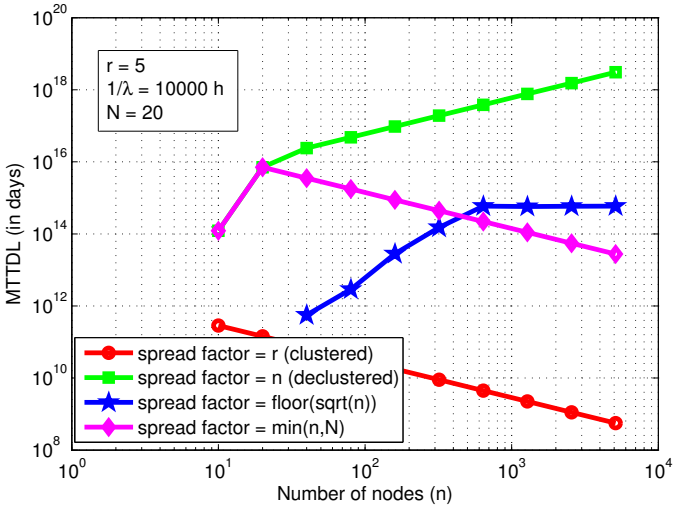

Fig. 5. Theoretical estimates of MTTDL for a replication factor of five with mean time to failure of a node equal to 10000 h.

amount of most-exposed data to be rebuilt as the system goes to higher exposure levels also decreases. It appears that, for declustered placement, the negative influence of limited network bandwidth is effectively countered by the positive influence of decreasing amounts of critical data as additional nodes fail.

Replication Factor 5**:** For declustered placement, the expression for MTTDL (24) reduces to

$$\text{MTTDL}^{\text{declus.}} = \begin{cases} \dfrac{\mu^4(n-1)^3(n-2)^2(n-3)}{768n\lambda^5} & \text{when } n \leq N+1 \\ \dfrac{\mu^4(N+1)^2N^3(N-1)}{768(N+2)\lambda^5} & \text{when } n = N+2 \\ \dfrac{\mu^4(N+2)^2(N+1)N^3}{768(N+3)\lambda^5} & \text{when } n = N+3 \\ \dfrac{\mu^4(n-1)^2(n-2)N^4}{768n(n-4)\lambda^5} & \text{when } n \geq N+4. \end{cases} \quad (31)$$

The above expressions are plotted in Fig. 5. The effect of decreasing amounts of critical data as additional nodes fail is stronger than that observed for replication factor four.

G. Optimal Data Placement

Using expression (24) for MTTDL, one can find the optimal value of the spread factor k for which the corresponding MTTDL is maximized. Clearly, the optimal spread factor depends on the number of nodes n and the maximum network rebuild bandwidth B_{max}. In a dynamically changing storage system, the number of nodes and the available network rebuild bandwidth B_{max} may change over time. As a result, the optimal spread factor may change as well. In this case, one could consider redistributing the data in accordance to the new optimal spread factor. Such a scheme ensures that the system reliability constantly remains at the highest level.

VI. Simulation

Event driven simulations similar to [2] are used to verify the theoretical estimates of MTTDL for different placement schemes. The storage system is simulated using an event-driven simulator with three types of events that drive the simulation time forward: *(a) failure events*, *(b) rebuild-complete events*, and *(c) node-restore events*. The state of the system is maintained by the following variables: `time`, the simulated time; `failTimes`, a list of times to next failure of each node drawn according to the chosen failure distribution; `failedNodes`, the list of nodes that have failed in the system and are being rebuilt; `exposureLevel`, the exposure level; and `dataExposure` $= (D_0, \cdots, D_r)$, a vector of length $(r+1)$ where D_l is the number of distinct data blocks that have lost l replicas. The values of these variables are updated at each event, and when $D_r > 0$, data loss is said to have occurred and the simulation ends. A network rebuild bandwidth constraint B is imposed during the simulation by limiting the total bandwidth of all rebuilds in the system to B. For each set of parameters, the simulation is run 100 times, and the MTTDL and its bootstrap 95% confidence intervals are computed.

Although some of the assumptions used in the theoretical analysis, such as independence of node failures, are also used in the simulation, the simulation results reflect a more realistic picture of the systems reliability. This is because of the following key differences between the theoretical analysis and the simulations. The theoretical estimate of MTTDL in (7) takes into account only the time spent by the system in the fully-operational mode and ignores the rebuild times, whereas the simulations do not ignore the rebuild times. Furthermore, in (9), P_{DL} is approximated by the probability of the direct path to data loss, thereby implicitly assuming that this is the only path following a first-node-failure event that would lead to data loss. In simulations however, all the complex trajectories of the system through the different exposure levels are simulated by

TABLE II
RANGE OF VALUES OF DIFFERENT SIMULATION PARAMETERS

Parameter	Meaning	Range
c	storage capacity of each node	12 TB
n	number of storage nodes	4 to 64
r	replication factor	2, 3, 4
b	rebuild bandwidth available at each node	96 MB/s
$1/\lambda$	mean time to failure of a node	300 to 10000 h
N	maximum number of nodes from which rebuild can occur at full speed in parallel	12
$B_{\max}$	maximum network rebuild bandwidth	1152 MB/s

simulating random node failure events and updating the data exposure vector by taking partial rebuilds into account. In the theoretical analysis, the time required to restore new nodes in a declustered placement scheme (following successful rebuild of lost replicas in the spare space of surviving nodes) is ignored, whereas in the simulations, the time to restore new nodes is simulated as well. In addition, other approximations made in the analysis, such as neglecting the effect of the transient period of the system, are implicitly avoided in the simulations. Therefore, the simulations reflect a more complex picture of the system behavior than that assumed in theory.

Table II shows the range of parameters used for the simulations. Typical values for practical systems are used for all parameters, except for the mean times to failure of a node, which have been chosen artificially low (10000 h and 1000 h for replication factors 2 and 3, respectively) to shorten the simulation times. The running times of simulations with practical values of the mean times to node failure, which are of the order of 10000 h or higher, are prohibitively high; this is due to the fact that P_{DL} becomes extremely low, thereby making the number of first-node-failure events that need to be simulated (along with the other complex set of events that restore all lost replicas following each first-node-failure event) extremely high for each run of the simulation. Although this approach scales down the MTTDL by making failure events more frequent, its use is justified because it preserves the ratios of MTTDLs of the various schemes [5].

The simulation-based values of MTTDL for replication factors 2 and 3 are plotted in Figs. 6 and 7, respectively. The simulations were done for the two extreme values of spread factors, namely, clustered placement ($k = r$) and declustered placement ($k = n$). It is seen that the simulation-based values are a good match to the theoretical estimates.

The following lists the findings of this work:

- Network rebuild bandwidth limitations affects the rebuild processes in a storage system significantly and decreases reliability.
- The decrease in reliability due to limited network rebuild bandwidth depends on the spread factor of the placement scheme used.
- For declustered placement, the decrease in MTTDL is proportional to n^{r-1}.
- For replication factor two and three, a placement scheme that limits the spread factor to the maximum number of nodes that the network rebuild bandwidth can support at full speed in distributed rebuild is observed to outperform the declustered placement scheme in MTTDL.
- For replication factors greater than three, limiting the spread factor does not yield any benefits over the declustered placement scheme as it did for replication factor three. This is because, the effect of decreasing amounts of most-exposed data at each exposure level is stronger than the effect of slower rebuilds.

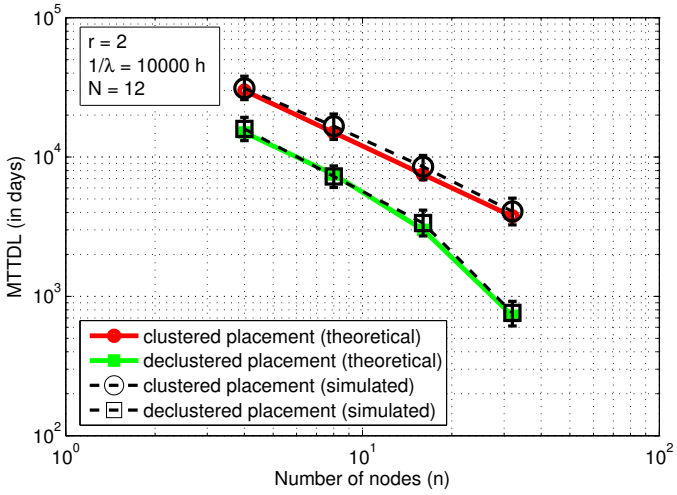

Fig. 6. Theoretical and simulation-based estimates of MTTDL for a replication factor of two with mean time to failure of a node equal to 10000 h. For the simulation-based values, 95% bootstrap confidence intervals are shown.

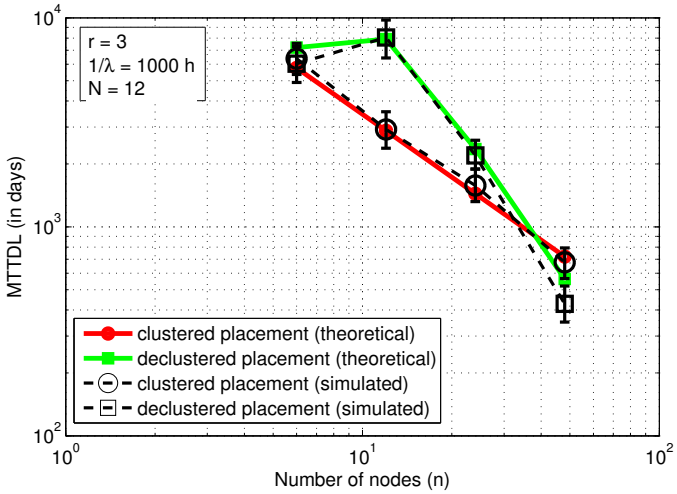

Fig. 7. Theoretical and simulation-based estimates of MTTDL for a replication factor of three with mean time to failure of a node equal to 1000 h. For the simulation-based values, 95% bootstrap confidence intervals are shown.

VII. CONCLUSIONS

We studied the effect of limited network rebuild bandwidth on the reliability of storage systems using theoretical analysis and simulations. It was shown that, if replicas are spread over a higher number of nodes than which the network rebuild bandwidth can support at full speed during a parallel rebuild

process, the system reliability is significantly reduced and a drastic change in MTTDL behavior is observed as the system size increases. Replicas are spread across many nodes in order to parallelize the rebuild process and improve system reliability; but when the network bandwidth is incapable of supporting such a high rate of data transfer across nodes during parallel rebuild, the rebuild times increase and the time window of vulnerability widens during which additional node failures can lead to data loss. For replication factors two and three, limiting the spread of replicas to the maximum number of nodes that the network rebuild bandwidth can support at full speed improves system reliability when compared to the declustered placement scheme where the replicas are spread across all the nodes. However, for higher replication factors, the declustered placement scheme outperforms such a scheme in MTTDL. This is attributed to the fact that spreading replicas over many nodes also decreases the amount of critical data that need to be rebuilt as additional nodes fail. Based on the results obtained, a method to maintain the highest reliability level in a dynamically changing storage environment was proposed. Extension of the present methodology to address reliability of systems using erasure codes is a subject of further investigation.

REFERENCES

[1] V. Venkatesan, I. Iliadis, X.-Y. Hu, R. Haas, and C. Fragouli, "Effect of replica placement on the reliability of large-scale data storage systems," in *Proc. 18th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'10)*, 2010, pp. 79–88.

[2] V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, "Reliability of clustered vs. declustered replica placement in data storage systems," in *Proc. 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'11)*, 2011, pp. 307–317.

[3] 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.

[4] 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.

[5] 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.

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

[7] V. Venkatesan and I. Iliadis, "A general reliability model for data storage systems," IBM Research - Zurich, Tech. Rep. RZ 3817, 2012.

[8] 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.

[9] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *Proc. 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)*, 2010, pp. 61–74.

[10] 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.

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

[12] 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, July 2011.