RZ 3733 (# 99743) 02/15/2009
Computer Science 11 pages

Research Report

Replication Versus RAID for Distributed Storage Systems

I. Iliadis, R. Haas

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

{ili,rha}@zurich.ibm.com

Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

Replication Versus RAID for Distributed Storage Systems

Ilias Iliadis, Robert Haas

IBM Zurich Research Laboratory
8803 Rüschlikon, Switzerland

{ili,rha}@zurich.ibm.com

ABSTRACT

In today's large-scale distributed storage systems, vast amounts of user data are stored among a large number of nodes and disks. High availability and increased reliability require that data be stored in a redundant manner. We consider the two popular redundancy schemes: replication and erasure coding. In particular, we consider RAID-type distributed storage systems. New analytical models are developed to assess the system reliability in terms of the mean time to data loss, the storage efficiency, and the I/O throughput performance. Furthermore, we address the issue of placement of the redundant data in the nodes, and examine its effect. The models are then extended to analytically assess the impact of unrecoverable or latent media errors encountered on disk drives. We propose to use the intradisk redundancy scheme to cope with those type of errors and enhance the reliability of the storage systems. Our analytical results show that distributed RAID-5 systems enhanced by the intradisk redundancy scheme provide improved reliability compared with mirroring replication systems. They also require less storage space, but incur I/O performance degradation.

1. 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, Freenet, Intermemory, OceanStore, CFS, and PAST, aim at providing inexpensive, highly-available storage without centralized servers (see [15] 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.

Replication is the simplest redundancy scheme where, r identical copies of each user data are kept at any instant in the system nodes. Wide-scale replication increases the reliability, availability, and durability, but it also increases the bandwidth and storage requirements of the system. The value of r should therefore be appropriately set to ensure the desired availability and performance levels.

Erasure codes provide sufficient redundancy, which is less than that of the replication schemes. Erasure codes divide a data entity into m fragments and recode them into n fragments, where $n > m$. The rate of encoding is then given by $r = m/n < 1$. A rate-r code increases the storage cost by a factor of $1/r$. The key property of maximum distance separable (MDS) erasure codes is that the original data can be reconstructed from any m of the n fragments. By storing each of the n fragments into a separate node, data is protected against the simultaneous failure of up to $n-m$ nodes. Such erasure codes are a superset of replication and RAID systems. For example, a system that creates four replicas for each block can be described by an $(m = 1, n = 4)$ erasure code. RAID level 1, 4, 5, and 6 systems can be described by $(m = 1, n = 2)$, $(m = 4, n = 5)$, $(m = 4, n = 5)$, and $(m = 4, n = 6)$ erasure codes, respectively.

The work in [15] quantifies the availability gained using erasure codes. It shows that erasure codes use an order of magnitude less bandwidth and storage than replication for systems with similar mean time to failure (MTTF). It also shows that employing erasure codes increases the MTTF of the system by many orders of magnitude over simple replication, with the same storage overhead and repair times.

Erasure coding in a dynamic error-prone environment requires the precise identification of failed or corrupted fragments. Without this ability, there potentially is a factorial number of $\binom{n}{m}$ combinations of fragments to be tested in order to reconstruct the original data. As a result, the system needs to detect whether a fragment has been corrupted, and, if so, it discards it. A secure verification hashing scheme can serve the dual purpose of identifying and verifying each fragment. As the original data can be reconstructed by any m correctly verified fragments, such a scheme is likely to increase the bandwidth and storage requirements. However, in [15] it is argued that the requirements are less than those corresponding to a replication system.

Another comparison of replication and erasure coding is presented in [11]. Unlike the comparison in [15], this paper considers the characteristics of the nodes, and concludes that in some cases the benefits from coding are limited, and may not be worth its disadvantages. Whereas previous comparisons mostly argue in favor of erasure coding, because of its huge storage savings for the same availability levels (or conversely, huge availability gains for the same storage levels), this work reaches a different conclusion. It argues that although gains from coding exist, they are highly dependent on the characteristics of the nodes that comprise the overlay network. In fact, the benefits of coding are so limited in some cases that they can easily be outweighed by some disadvantages and the additional complexity of erasure codes. In particular, the savings of erasure coding are higher when data is stored in unreliable servers (lower server availability levels) or when the reliability guarantees are more stringent (higher number of nines in data availability). The redundancy gains from using erasure coding range from 1- to 3-fold. Clearly, erasure coding prevails when we consider the redundancy savings and the smaller amount of data that need to be written. But the authors in [11] argue that more important than these two aspects is the savings in the bandwidth required to restore redundancy levels in the presence of a changing membership. This stems from the fact that bandwidth, and not spare storage, is most likely the limiting factor for the scalability of peer-to-peer storage systems.

The main point raised against the use of coding is that it introduces complexity in the system. Not only is there complexity associated with the encoding and decoding of the blocks, but the entire system design becomes more complex (e.g., the task of redundancy maintenance becomes more complicated). Another point against the use of erasure codes is the download latency in a environment like the Internet where the inter-node latency is very heterogeneous. When using replication, data can be downloaded from the replica that is closest to the client, whereas with coding the download latency is bounded by the distance to the m-th closest replica. Coding also complicates the task of downloading only a particular subset of the data object (a sub-block), as the entire data object must be reconstructed. With full replicas, sub-blocks can be downloaded trivially. A similar observation is that erasure coding is not adequate for a system design in which operations are done at the server side, like keyword searching.

Regarding erasure codes, recent research suggests that low-density parity-check (LDPC) codes can achieve high coding bandwidth and near-optimal coding efficiency. Some near-optimal erasure codes, such as Luby-transform (LT) codes, allow data reconstruction with a significant higher read flexibility than plain replication. These codes are rateless in that they can generate a practically infinite number of coded blocks. They also achieve a good tradeoff between transmission overhead and computation overhead. RobuSTore [16] uses speculative access to exploit the rateless feature of these erasure codes. Combining erasure codes and speculative access leads to increased performance. It is also argued that an erasure-code-based scheme is more effective in adapting to performance variation than a replication-based scheme.

From the preceding, it follows that each of the two schemes has advantages and disadvantages, and therefore additional work is needed to further investigate their impact on system performance in depth.

The key contributions of this paper are the following. We consider the replication and RAID-type distributed storage systems. New analytical models are developed to assess the system reliability in terms of the mean time to data loss (MTTDL), the storage efficiency, and the I/O throughput performance. Furthermore, we address the issue of placement of the redundant data in the nodes, and examine its effect. Our analysis shows that the reliability is insensitive to the way replication data are placed in the nodes. The models are then extended to analytically assess the impact of unrecoverable or latent media errors encountered on disk drives. As our results demonstrate, the presence of unrecoverable errors decreases the reliability level by orders of magnitude. The results obtained also reveal that, contrary to previous conviction, the reliability of a mirroring replication system can be lower than that of a distributed RAID-5 system, when the probability of unrecoverable sector errors is sufficiently high. We therefore propose to use the intradisk redundancy scheme to cope with those type of errors, and enhance the reliability of the storage systems, especially in the presence of multiple correlated media errors on the same track or cylinder of the hard-disk drives (HDDs). Our analytical results show that distributed RAID-5 systems enhanced by the intradisk redundancy scheme provide improved reliability compared with mirroring replication systems. They also require less storage space, but incur I/O performance degradation.

The remainder of the paper is organized as follows. Section 2 discusses results obtained by previous works considering the replication scheme. Section 3 describes the replication and distributed RAID schemes, introduces the relevant performance measures, and addresses the placement issue of redundant data in the system nodes. Also, closed-form expressions for the MTTDL of the two schemes are derived. Numerical results demonstrating their relative reliability are provided in Section 4. Section 5 considers the issue of unrecoverable or latent errors and reviews the extent to which these errors occur. The basic intra-disk redundancy scheme developed for increasing the reliability of disks in the presence of unrecoverable errors and disk failures is briefly reviewed in Section 6. In Section 7, the I/O performance of the various systems is considered in terms of the saturation throughput that is evaluated analytically. Section 8 presents numerical results demonstrating the effectiveness of the mirroring replication and distributed RAID-5 schemes in both variants, i.e., plain as well as enhanced by the intra-disk redundancy scheme. Finally, we conclude in Section 9.

2. RELATED WORK

2.1 Replication

The problem of using replication to reliably maintain state in a distributed system for time spans that far exceed the lifetimes of individual replicas is addressed in [10]. This scenario is relevant for any system comprised of a potentially large and selectable number of replicated components, each of which may be highly unreliable, where the goal is to have enough replicas to keep the system "alive" (meaning at least one replica is working or available) for a certain expected period of time, i.e., the system's lifetime. In particular, this applies to recent efforts to build highly available storage systems based on the peer-to-peer paradigm. This work studies the impact of practical considerations, such as storage and bandwidth limits, on the system and presents methods to optimally choose system parameters so as to maximize lifetime. The analysis presented reveals that, for most practical scenarios, it is better to invest the available repair bandwidth in aggressively maintaining a small number of replicas than spreading it across a large number of replicas.

2.2 Replication Strategies

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 performance and reliability [8]. This work presents a comparative analysis of replication algorithms that are based upon a specific design of DHT. It also presents a novel maintenance algorithm for dynamic replica placement, and considers the reliability of the resulting designs at the system level.

According to this approach, replicas of a data item are placed on the r successors of the node responsible for that item's key. In the case of dynamic replication, replica placement is performed based on an allocation function. Random placement is not considered to be a realistic option because it would lead to high maintenance costs and make it impossible to exploit local routing information. This work 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. It is, however, stated that this scheme has the highest overhead, because of discontinuities in the placement function.

3. REPLICATION VERSUS DISTRIBUTED RAID

We consider data stored in nodes that are subject to failure. Our goal is to increase the system reliability and data availability in the presence of node failures. Here, we consider the case where the expected time of data availability is orders of magnitude higher than that of the node availability, such that data is lost only because of node failures. We consider the following two schemes for increasing system reliability:

Replication Scheme: Data D_i stored in a node is replicated r times, i.e., it is also stored in $r-1$ additional nodes. In this case data D_i is lost when all of the r nodes fail.

Distributed RAID Scheme: Data D_i is divided into m pieces, $D_{i,1}, \ldots, D_{i,m}$, and stored in m different nodes. Furthermore, additional parity information is generated in a RAID fashion and stored in additional nodes.

Let us make the following assumptions:

(A1) Node failure rates are independent and exponentially distributed with parameter $\lambda = 1/\mathrm{MTTF}_n$.

(A2) The rebuild time R_i for data D_i follows an exponential distribution with parameter $\mu_i = 1/\overline{R}_i$.

(A3) The rebuild time R for all data of a node follows an exponential distribution with parameter $\mu = 1/\overline{R}$.

Note that the above assumptions refer to nodes, not to disks. As disk failures do not necessarily imply node failures, node failures can be independent with exponentially distributed rates, in contrast to the disk failures that are neither independent nor exponentially distributed [13].

We now proceed by addressing the following question. How should the data be stored in the available nodes to maximize the MTTDL? For the purposes of our discussion, let us start by considering the case of replication with $r = 2$. Let $D_1, D_2, \ldots, D_k$ represent the user data stored in a node, say node n_0, and $D'_1, D'_2, \ldots, D'_k$ the replicated data stored in nodes $n_1, n_2, \ldots, n_q$, respectively, with the nodes not all being necessarily different. Without loss of generality, we also assume that the user data are of equal size, and that the system is homogeneous, such that the corresponding rebuild times $R_1, R_2, \ldots, R_k$ are identically distributed. From assumption (A2), it follows that $\mu_1 = \cdots = \mu_i = \cdots = \mu_k$, with $1 \le i \le k$. When node n_0 fails, the system enters the *critical mode* and starts the rebuild of the data from the replicas. The mode is critical because if one of nodes $n_1, n_2, \ldots, n_q$ fails, this would lead to data loss. Let F denote the time to the next node failure while in critical mode. From assumption (A1), it follows that F is exponentially distributed with parameter $\lambda^{(Q)} = Q\lambda$, where Q is the number of different nodes in the set $\{n_1, n_2, \ldots, n_q\}$. Clearly, F is maximized when $Q = 1$, that is, when $n_1 = n_2 = \cdots = n_q$, which implies that all replicas are stored in the same node. This is in agreement with the discussion in Section 2.2, where it is argued that the probability of data loss is minimized when data are placed together rather than being spread. But according to the discussion in Section 7.1 of [3] [1], the duration of the critical period also depends on the rebuild time R, and is equal to the minimum of F and R. From assumption (A3), it follows that R is exponentially distributed with a parameter $\mu^{(Q)}$ that depends on Q. The probability P_{fr} that the critical mode ends because of another node failure is then given by

$$P_{\mathrm{fr}} \;=\; P(F < R) = \frac{\lambda^{(Q)}}{\lambda^{(Q)} + \mu^{(Q)}}. \qquad (1)$$

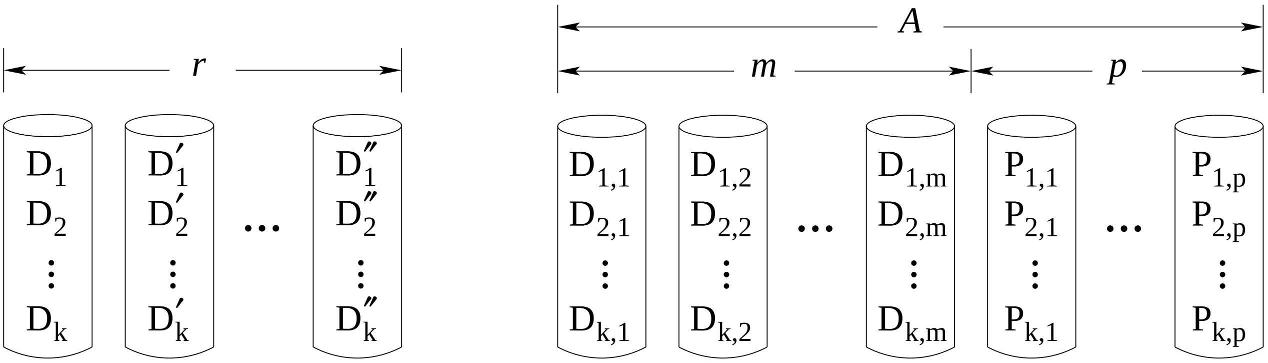
The MTTDL of the system increases when P_{fr} decreases. Note that minimizing $\lambda^{(Q)}$, by placing all replicas in the same node ($Q = 1$), results in an increase of the rebuild time, because all data need to be read from the same node. Therefore, as the parameter $\mu^{(Q)}$, which is given by μ_i/k, is also minimized, it is not evident whether P_{fr} is minimized. From (1) it follows that for $Q = 1$, $P_{\mathrm{fr}} = \lambda/(\lambda + \mu_i/k)$. On the other hand, if all replication data are placed in k different nodes, that is $Q = k$, then F is minimized, with $\lambda^{(Q)} = k\lambda$. Assuming that the rebuild is performed on k different nodes, the rebuild time is also minimized with $\mu^{(Q)} = \mu_i$. In this case $P_{\mathrm{fr}} = k\lambda/(k\lambda + \mu_i)$, which is the same as the one derived above in the case of $Q = 1$. Consequently, the reliability does not seem to be affected by the node placement of the replication data. Based on that, we will consider in the remainder of the paper that the redundancy data are placed together. This means that in the general case of replication, $D_1, D_2, \ldots, D_k$ are stored in one node, and the replicated data are stored in the same $r-1$ additional nodes, as shown in Figure 1(a).

In the case of distributed RAID with a number of p parities, the corresponding array size A is equal to $m + p$. Thus, $D_{1,1}, \ldots, D_{1,m}$ and $P_{1,1}, \ldots, P_{1,p}$ are stored in A different nodes, as shown in Figure 1(b). Subsequent data are stored in the same nodes, such that $D_{1,1}, D_{2,1}, \ldots, D_{k,1}$ are stored in one node, $D_{1,m}, D_{2,m}, \ldots, D_{k,m}$ in another node, $P_{1,1}, P_{2,1}, \ldots, P_{k,1}$ in another node, and $P_{1,p}, P_{2,p}, \ldots, P_{k,p}$ in another node. With this arrangement, in every group of $A - p$ nodes containing data corresponds a group of p nodes containing parities.

The notation used for the analysis is given in Table 1. The parameters are divided into two sets, namely, the set of independent and that of dependent parameters, listed in the upper and lower part of the table, respectively. It is also assumed that the node failure rate is much higher than the disk failure rates.

From the above assumptions, it follows that the probability P_f that at an arbitrary time the contents of a tagged

[1] Note that the formulas derived in that work for the case of disk failures can also be applied to obtain results for the case of node failures, as done in this work.



(a) Replication scheme: replication group.

(b) Distributed RAID scheme: array group.

Figure 1: Distributed redundancy schemes.

Parameter	Definition
$1/\lambda$	Mean time to failure for a node
$1/\mu$	Mean time to rebuild for a node
N_{repl}	Number of nodes in the system in the case of replication
N_{RAID}	Number of nodes in the system in the case of distributed RAID
g_{repl}	Number of replication groups in the system in the case of replication
g_{RAID}	Number of array groups in the system in the case of distributed RAID
r	Replication factor
A	Number of nodes per array group
p	Number of parity nodes per array group
MTTF_n	Mean time to failure for a node
MTTR_n	Mean time to recover the contents of a failed node
P_f	Probability that at an arbitrary time the contents of a tagged node are being rebuilt because the node has failed
$\text{MTTDL}_{\text{group}}$	Mean time to data loss for a replication group
$\text{MTTDL}_{\text{array}}$	Mean time to data loss for an array group
$\text{MTTDL}_{\text{replication}}$	Mean time to data loss for the entire replication system
$\text{MTTDL}_{\text{RAID}}$	Mean time to data loss for the entire distributed RAID system
$se^{(\text{replication})}$	Storage efficiency of the replication scheme
$se^{(\text{RAID})}$	Storage efficiency of the RAID scheme
s	Storage-requirement factor of the space required by the distributed RAID approach compared with the space required by the replication approach

Table 1: Notation of system parameters.

node are being rebuilt because the node has failed is given by

$$P_f = \frac{\lambda}{\lambda + \mu} \; . \tag{2}$$

It also holds that

$$\text{MTTDL}_{\text{replication}} \;=\; \frac{\text{MTTDL}_{\text{group}}}{g_{\text{repl}}} \; , \tag{3}$$

and

$$\text{MTTDL}_{\text{RAID}} \;=\; \frac{\text{MTTDL}_{\text{array}}}{g_{\text{RAID}}} \; . \tag{4}$$

The storage efficiency of a replication system is given by

$$se^{(\text{replication})} \;=\; \frac{1}{r} \; . \tag{5}$$

Assuming that the replication system is comprised of N_{repl} nodes, the number of nodes containing user data (not replicated data) is equal to N_{repl}/r.

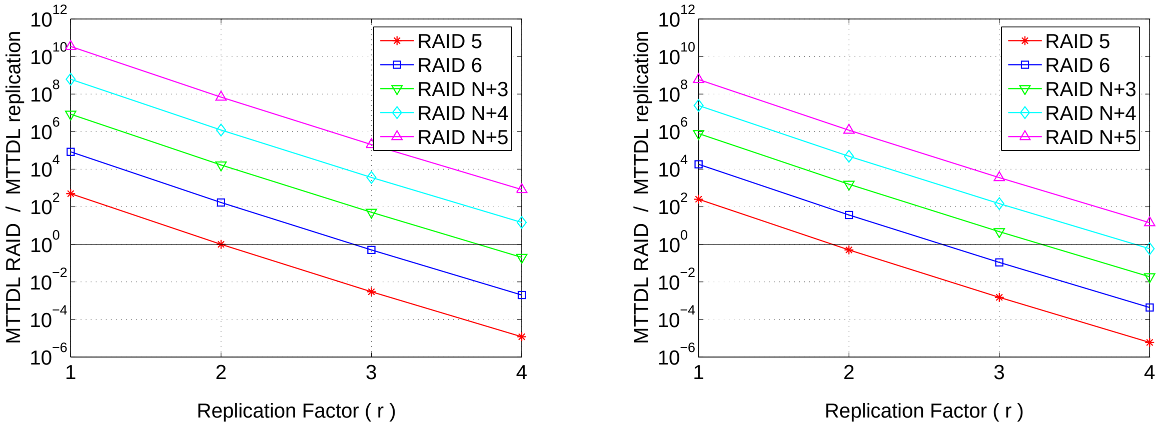
The storage efficiency of a distributed RAID system is given by

$$se^{(\text{RAID})} \;=\; \frac{A-p}{A} \;=\; 1 - \frac{p}{A} \; . \tag{6}$$

Assuming that the system is comprised of N_{RAID} nodes, the number of nodes containing user data (not parity data) is equal to $N_{\text{RAID}}(A-p)/A$. Considering a replication system and a RAID distributed system storing the same amount of user data, yields $N_{\text{repl}}/r = N_{\text{RAID}}(A-p)/A$. Consequently, the storage-requirement factor s of the space required by the distributed RAID approach compared with the space required by the replication approach is given by

$$s \;\triangleq\; \frac{N_{\text{RAID}}}{N_{\text{repl}}} = \frac{se^{(\text{replication})}}{se^{(\text{RAID})}} = \frac{A}{r(A-p)} \; . \tag{7}$$

Also, the number of replication groups and the number of



(a) $A(1) = A(\text{RAID 5}) = 2$.

(b) $A(1) = A(\text{RAID 5}) = 4$.

Figure 2: Ratio of MTTDL of distributed RAID to MTTDL of replication.

RAID array groups in the system are given by

$$g_{\text{repl}} = \frac{N_{\text{repl}}}{r} \qquad \text{and} \qquad g_{\text{RAID}} = \frac{N_{\text{RAID}}}{A} \,, \tag{8}$$

respectively.

In the case of distributed RAID, data in a given array group is lost when there are $p+1$ simultaneous node failures. In particular, for $\lambda \ll \mu$, it can be shown that

$$\text{MTTDL}_{\text{array}} \approx \frac{\mu^p}{A(A-1)\cdots(A-p)\,\lambda^{p+1}} \,, \tag{9}$$

which is the extension of Eqs. (46) and (53) of [3] derived for the cases of $p = 1$ (RAID-5 system) and $p = 2$ (RAID-6 system), respectively.

In the case of replication, data in a given replication group is lost when all r nodes fail. The corresponding MTTDL is obtained from (9) by substituting $A = r$ and $p = r - 1$. Thus,

$$\text{MTTDL}_{\text{group}} \approx \frac{\mu^{r-1}}{r!\,\lambda^r} \,. \tag{10}$$

From (3) to (10), it follows that the ratio $f(r,p)$ of the MTTDLs corresponding to the distributed RAID and replication schemes is given, as a function of r and p, by

$$\begin{aligned} f(r,p) &\triangleq \frac{\text{MTTDL}_{\text{RAID}}}{\text{MTTDL}_{\text{replication}}} = \frac{\text{MTTDL}_{\text{array}}\, g_{\text{repl}}}{\text{MTTDL}_{\text{group}}\, g_{\text{RAID}}} \\ &= (A-p)\,\frac{\text{MTTDL}_{\text{array}}}{\text{MTTDL}_{\text{group}}} \\ &= \frac{r!}{A(A-1)\cdots[A-(p-1)]} \left(\frac{\lambda}{\mu}\right)^{r-p-1} \\ &= \frac{r!\,\rho^{r-p-1}}{A(A-1)\cdots[A-(p-1)]} \,, \end{aligned} \tag{11}$$

where

$$\rho \triangleq \frac{\lambda}{\mu} \,. \tag{12}$$

Combining (2) and (12), (11) yields

$$\begin{aligned} f(r,p) &\triangleq \frac{\text{MTTDL}_{\text{RAID}}}{\text{MTTDL}_{\text{replication}}} \\ &= \frac{r!}{A(A-1)\cdots[A-(p-1)]} \left(\frac{P_f}{1-P_f}\right)^{r-p-1} \,. \end{aligned} \tag{13}$$

A(1) \ **r**	1	2	3	4
2	2	1	0.666	0.5
4	1.333	0.666	0.444	0.333

Table 2: Storage-requirement factor s of distributed RAID versus replication systems.

4. RELIABILITY RESULTS

We also consider replication systems with $r = 1, 2, 3,$ and 4. We consider RAID-5, RAID-6, RAID-(N+3), RAID-(N+4), and RAID-(N+5) type of systems, which correspond to $p = 1, 2, 3, 4,$ and 5, respectively. The erasure code overhead is assumed to be fixed by taking the array length to be proportional to p, i.e., $A(p) = p\,A(1)$, where $A(1)$ denotes the array length in a RAID-5 configuration. We also set $P_f = 10^{-3}$.

Table 2 lists the storage-requirement factor s, given by (7), as a function of $A(1)$ and r. We observe that as $A(1)$ and r increase, the factor s decreases. Note that a RAID-5 configuration with $A(1) = 2$ is equivalent to a replication system with $r = 2$, and therefore $s = 1$.

Figure 2 shows the ratio of the MTTDL for distributed RAID to the MTTDL for replication as a function of r, for various RAID configurations having the same storage efficiency as a RAID-5 system with array sizes of $A = 2$ and $A = 4$. We observe that as r increases, the MTTDL ratio decreases, but the storage-requirement factor s also decreases. Also, as p increases, the system reliability for distributed RAID increases. For $A = 2$, increasing p by one results in an MTTDL increase by roughly two orders of magnitude.

From Figure 2(a), we note that a distributed RAID-6 system with an array size equal to 4 and $p = 2$ provides roughly the same reliability as that of a replication system with $r = 3$, but it requires a storage space which is only 66% of that required by the replication system. From Figure 2(b), we note that a distributed RAID-(N+4) system with an array size equal to 16 and $p = 4$ provides roughly the same reliability as that of a replication system with $r = 3$, but requires a storage space which is only 33% of that required by the replication system.

Parameter	Definition
D_n	Number of disks per node
C_d	Disk drive capacity
S	Sector size
ℓ	Number of sectors in a segment
m	Number of parity sectors in a segment or number of interleaves or interleaving depth
P_{bit}	Probability of an unrecoverable bit error (data sheet specification)
P_{sec}	Probability of an unrecoverable sector error (data sheet specification)
C_n	Node capacity
$se^{(\text{RAID})}$	Storage efficiency of the RAID scheme
$se^{(\text{IDR})}$	Storage efficiency of the intra-disk redundancy (IDR) scheme
$se^{(\text{replication+IDR})}$	Storage efficiency of the RAID system enhanced by IDR
$se^{(\text{RAID+IDR})}$	Storage efficiency of the replication system enhanced by IDR
P_s	Probability of an unrecoverable error on a tagged sector at an arbitrary time

Table 3: Notation of system parameters.

5. UNRECOVERABLE ERRORS

The reliability of storage systems is also degraded by the occurrence of unrecoverable or latent sector errors, that is, of errors that cannot be corrected by either the standard sector-associated error-correcting code (ECC) or the re-read mechanism of the HDDs. As the occurrence of unrecoverable or latent sector errors and therefore the percentage of drives that develop such errors increase with disk capacity [5, 9], the emergence of high-capacity SATA drives as a low-cost alternative to SCSI and FC drives in data storage systems has brought the issue of system reliability to the forefront.

Techniques such as disk scrubbing [12, 14] and intradisk redundancy [2, 3] have been proposed to enhance reliability. The established, widely used disk scrubbing scheme periodically accesses disks to detect media-related unrecoverable errors. The scrubbing process identifies unrecoverable sector errors at an early stage and attempts to recover them. Thus, the scrubbing effectively reduces the probability of encountering unrecoverable sector errors when a failure occurs. On the other hand, the recently proposed intra-disk redundancy scheme uses a further level of redundancy inside each disk. It is based on an interleaved parity-check coding scheme [3], which incurs only negligible I/O performance degradation and has been developed to increase the reliability of disks in general, but especially in the presence of multiple correlated media errors on the same track or cylinder.

A thorough comparison of these two schemes was presented in [6]. It was demonstrated that the reliability improvement due to disk scrubbing depends on the scrubbing frequency and the workload of the system, and may not reach the reliability level achieved by the intra-disk redundancy scheme, which is insensitive to the workload. For this reason, we consider the latter scheme in the remainder of the paper.

The parameters associated with the intra-disk redundancy scheme were chosen in such a way as to ensure sufficient degrees of storage efficiency, I/O performance, and reliability [3]. It was demonstrated that, for SATA disk drives, a RAID-5 system enhanced by the intra-disk redundancy scheme achieves a similar reliability as that of a RAID-6 system. The parameter choice was based on the assumption that the unrecoverable sector error probability is the one listed in the data sheet specifications provided by the disk manufacturers. However, empirical field results reported recently [1] suggest that the actual values can be orders of magnitude higher than the values previously assumed [7].

Next we study the reliability of the replication and RAID systems, in terms of the MTTDL, and find that the reliability level is adversely affected by the presence of unrecoverable or latent errors. We then demonstrate that the reliability level can significantly be improved by enhancing these systems with the intra-disk redundancy scheme. The notation used for our analysis is given in Table 3. The parameters are divided into two sets, namely, the set of independent and that of dependent parameters, listed in the upper and lower part of the table, respectively.

According to data sheet specifications, the likelihood of unrecoverable errors occurring in SATA drives is ten times higher than that in SCSI/FC drives [5]. The unrecoverable bit error probability P_{bit} is estimated to be 10^{-15} for SCSI and 10^{-14} for SATA drives. For a sector size of 512 bytes (the default for nearline disks), the equivalent unrecoverable sector error probability is $P_{\text{sec}} \approx P_{\text{bit}} \times 4096$, which is $4.096{\times}10^{-12}$ in the case of SCSI and $4.096{\times}10^{-11}$ in the case of SATA drives. In practice, however, and based on the empirical field results recently reported in [1], this probability seems to be much higher. In fact, it can be as high as $5{\times}10^{-9}$ [7], which is more than two orders of magnitude higher than the data sheet specifications for SATA and SCSI/FC disk drives. This, in turn, suggests that the reliability of SATA drives should be studied for values of the unrecoverable sector error probability in the range $[4.096{\times}10^{-11}, 5{\times}10^{-9}]$ rather than only for the data sheet specification value of $4.096{\times}10^{-11}$. As we will see in Section 8, increasing the probability of unrecoverable sector errors in this wide range has a significant impact on the system reliability.

6. INTRA-DISK REDUNDANCY SCHEME

Here we briefly review the intra-disk redundancy (IDR) scheme presented in [2] and developed to increase the reliability of disks in general, but especially to cope with the adverse effect of the spatial locality of errors, such as correlated media errors on the same track or cylinder of a disk [1]. A number of n contiguous data sectors in a strip as well as m redundant sectors derived from these data sectors are grouped together, forming a segment. The redundant parity sectors are obtained using a simple XOR-based interleaved parity-check (IPC) coding scheme [3], which, for small unrecoverable sector error probabilities not exceeding 10^{-8}, is shown to be as effective as the optimum, albeit more complex, Reed–Solomon (RS) coding scheme. The entire segment, comprising ℓ data and parity sectors, is stored contiguously on the same disk, where $\ell = n + m$. Note that this scheme addresses the issue of spatial locality of errors in that it can correct a single burst of m consecutive sector errors occurring in a segment. However, unlike the RS scheme, it in general does not have the capability of correcting any m sector errors in a segment.

The segment size ℓ and the number m of parity sectors in a segment are design parameters that can be optimized based on the desired set of operating conditions such that sufficient degrees of storage efficiency, performance and reliability are ensured [7]. In general, more redundancy (larger m) provides better protection against unrecoverable media errors. However, it also incurs more overhead in terms of storage space, computations required to obtain and update the parity sectors, and I/O operations. Furthermore, for a fixed degree of storage efficiency, increasing the segment size results in an increased reliability, but also in an increased penalty on the I/O performance. Therefore, a judicious trade-off between these competing requirements needs to be made. The storage efficiency $se^{(\mathrm{IDR})}$ of the IDR scheme is given by

$$se^{(\mathrm{IDR})} = \frac{\ell - m}{\ell} = 1 - \frac{m}{\ell} . \qquad (14)$$

A reasonable choice for the size of a segment and the number of parity sectors in a segment is $\ell = 128$ and $m = 8$, respectively [3]. The storage efficiency $se^{(\mathrm{IDR})}$ of the IDR scheme is then equal to 94%. The choice of $m = 8$ seems to be reasonable given that recent empirical data indicate that the median number of errors for disks containing one or several errors is 3 [1].

The overall storage efficiency of a replication system enhanced by the intra-disk redundancy scheme is then given by

$$\begin{aligned} se^{(\mathrm{replication+IDR})} &= se^{(\mathrm{replication})}\, se^{(\mathrm{IDR})} \\ &= \frac{1}{r}\left(1 - \frac{m}{\ell}\right) . \end{aligned} \qquad (15)$$

Similarly, the overall storage efficiency of a RAID-5 system enhanced by the intra-disk redundancy scheme is given by

$$\begin{aligned} se^{(\mathrm{RAID+IDR})} &= se^{(\mathrm{RAID})}\, se^{(\mathrm{IDR})} \\ &= \left(1 - \frac{p}{A}\right)\left(1 - \frac{m}{\ell}\right) . \end{aligned} \qquad (16)$$

7. I/O PERFORMANCE ANALYSIS

Here we assess the saturation throughput of the various schemes. In particular, we evaluate the I/O equivalent metric IOE, because the saturation throughput of a RAID system is inversely proportional to this metric [7]. The two key components that make up the time required for the processing of an I/O request to a disk are the seek time and the access time. The seek time depends on the current and the desired position of the disk head and is typically specified using an average value corresponding to a seek that requires the head to move half of the maximum possible movement. The access time depends on the size of the data unit requested. The processing time is determined by the type of workload (e.g., random vs. sequential I/O) and the size of the data unit. The processing time of an I/O request normalized to the seek time is expressed by the IOE, which was introduced in [4], where it is shown that the IOE of an I/O request containing k 4-KB chunks is given by

$$IOE = 1 + k/50 . \qquad (17)$$

For RAID-5 arrays, writing small (e.g., 4 KB) chunks of data located randomly on the disks poses a challenge, the so-called "small-write" problem. This is because each write operation to data also requires the corresponding RAID parity to be updated. A practical way to do this is to read the old data and the old parity from the two corresponding disks, compute the new parity, and then write the new data and the new parity. Hence, each small-write request results in four I/O requests being issued. Because of the small size of the data units involved, the predominant component of the processing time for each I/O request is the seek time. In [7] it is shown that the corresponding normalized time required for the processing of a small-write request for RAID 5, expressed through the IOE metric, is given by $4(1 + n/400)$, where n is the I/O request size expressed in sectors. In the case of a RAID-5 array comprised of two disks, which equivalently is a mirroring replication system, each small-write request results in two I/O requests being issued. Thus, the IOE metric is given by

$$IOE(n) = \begin{cases} 2\,(1 + n/400) & \text{for } A = 2 \\ 4\,(1 + n/400) & \text{for } A > 2 . \end{cases} \qquad (18)$$

Using the intra-disk redundancy scheme requires that the intra-disk parity must also be updated whenever a data unit is written. For a small write, a practical solution is to read the old data and the corresponding old intra-disk parity as part of a single I/O request. Then the new data and the new intra-disk parity are computed and subsequently written back to the disk by a single I/O request. The size of the requested data increases, thereby increasing the access time. However, for small writes and an appropriately designed IDR scheme, the processing time is still dominated by the seek time. Extending Equation (13) of [7] for the case of $A = 2$, the corresponding IOE metric for RAID 5 is obtained by

$$IOE = \begin{cases} 2\,(1 + \bar{n}/400) & \text{for } A = 2 \\ 4\,(1 + \bar{n}/400) & \text{for } A > 2 , \end{cases} \qquad (19)$$

where $\bar{n}$ is the average length of a single-sector write request when the IDR scheme is used is given by

$$\bar{n} = \begin{cases} 1 + \dfrac{\ell^2}{4(\ell - m)} & \text{for } \ell/m \text{ even} \\ 1 + \dfrac{\ell + m}{4} & \text{for } \ell/m \text{ odd.} \end{cases} \qquad (20)$$

8. NUMERICAL RESULTS

Here we analytically assess the effectiveness of the mirroring replication and distributed RAID-5 schemes. It is assumed that each node contains 15 300GB SATA disks, such that $D_n = 15$, $C_d = 300$ GB, and $C_n = D_n C_d = 4.5$ TB.

First, we consider the reliability of a replication system with a replication factor of $r = 2$ in the absence of unrecoverable sector errors. The system reliability is assessed

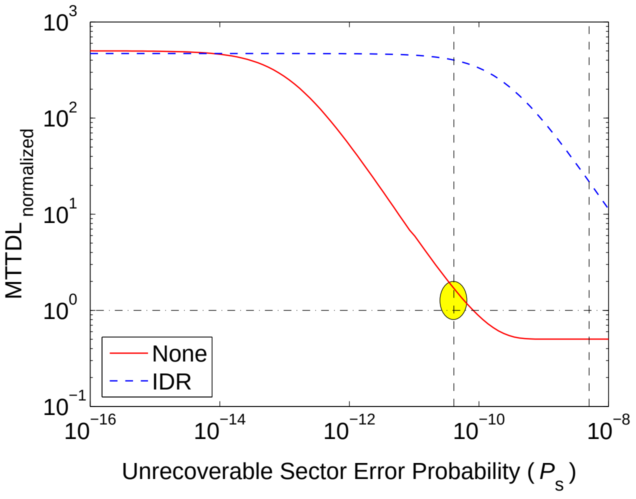


Figure 3: Normalized MTTDL of a replication system with $r = 2$ and $P_f = 10^{-3}$ under correlated unrecoverable sector errors ($\ell = 128$, $m = 8$).

in terms of the MTTDL. In Section 3, it was argued that the MTTDL does not depend on the method according to which redundant data is placed on the nodes. We therefore proceed by assuming that all redundant data corresponding to user data contained in a node is placed on the same node. This results in a RAID-10 mirroring replication system. Let $\text{MTTF}_{\text{n,s}}$ be the mean time to failure of the first node containing user data. As there are g_{repl} replication groups containing user data, and based on assumption (A1), it holds that

$$\text{MTTF}_{\text{n,s}} = \frac{\text{MTTF}_n}{g_{\text{repl}}} \tag{21}$$

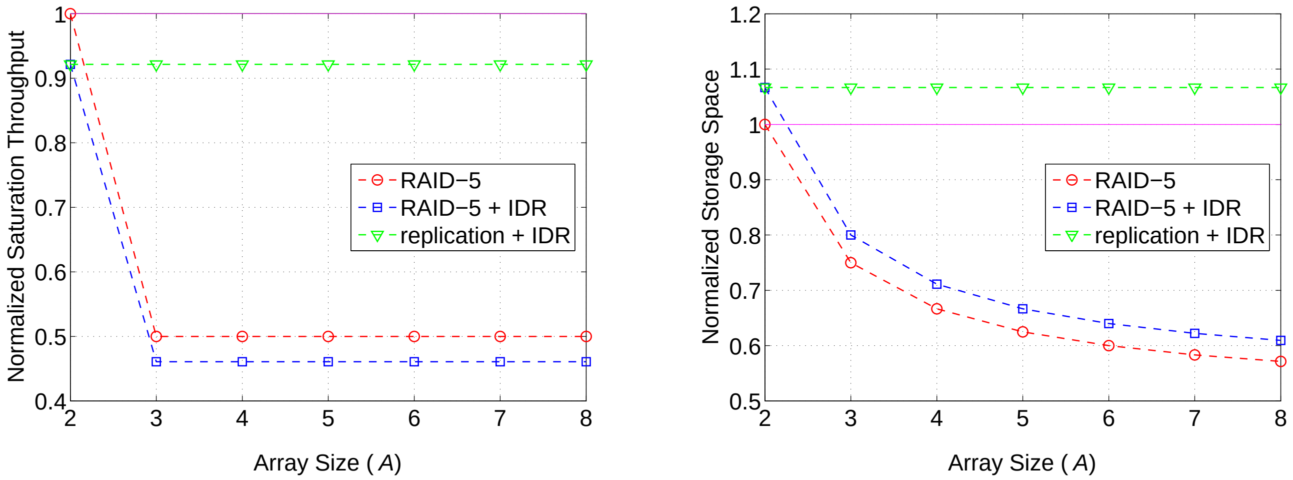
Combining (3) and (21) yields

$$\frac{\text{MTTDL}_{\text{replication}}}{\text{MTTF}_{\text{n,s}}} = \frac{\text{MTTDL}_{\text{group}}}{\text{MTTF}_n} \; . \tag{22}$$

The preceding reveals that the MTTDL of the system normalized to $\text{MTTF}_{\text{n,s}}$ is independent of the system size, and is equal to $\text{MTTDL}_{\text{normalized}}$, the MTTDL of a replication group normalized to the mean time to node failure, MTTF_n. We proceed by evaluating $\text{MTTDL}_{\text{normalized}}$, which can be analytically obtained using Equation (45) in [3] by substituting $N = 2$, $\lambda = 1$, $C_d = C_n$. Also, from (2) it follows that $\mu = \lambda P_f/(1 - P_f)$. The $\text{MTTDL}_{\text{normalized}}$ corresponding to $P_f = 10^{-3}$ is shown in Figure 3 as a function of the unrecoverable sector error probability. The interval $[4.096{\times}10^{-11}, 5{\times}10^{-9}]$ of practical importance for P_s is indicated between the two vertical dashed lines. In particular, the left vertical dashed line indicates the SATA drive specification for unrecoverable sector errors. Note that for small sector error probabilities, the MTTDL remains unaffected because data is lost owing to a node rather than an unrecoverable failure. In particular, the MTTDL of the replication group is 501 times MTTF_n, which is more than two orders of magnitude higher than MTTF_n. However, as the sector error probability increases, the MTTDL decreases. For $P_s = 4.096{\times}10^{-11}$ the MTTDL is equal to 1.69 times MTTF_n, which is of the same order as the mean time to a node failure (indicated by the ellipse). Thus, the presence of unrecoverable errors reduces the system reliability by almost three orders of magnitude. The MTTDL decrease ends when the sector error probability is larger than $5{\times}10^{-10}$, in which case the rebuild process in critical mode cannot be successfully completed because of systematic unrecoverable failures. In this case, the MTTDL is the mean time until the group (i.e., any one of the two nodes) enters the critical mode, which occurs after an expected time of $\text{MTTF}_n/2$, resulting in a normalized MTTDL of 0.5. The IPC-based IDR scheme, however, improves the MTTDL significantly. In particular, for $P_s = 4.096{\times}10^{-11}$, the corresponding normalized MTTDL is 401. Consequently, the IDR scheme improves the MTTDL by more than two orders of magnitude, therefore eliminating the negative impact of the unrecoverable sector errors.

Next we consider the reliability of mirroring replication ($r = 2$) and distributed RAID-5 systems. We start by assessing the saturation throughput achieved by the various systems in the case of small single-sector writes. From the discussion in Section 7, it follows that the maximum saturation throughput is achieved by the replication system. As the saturation throughput of a RAID system is inversely proportional to the IOE metric [7], the saturation throughput of the distributed RAID-5 system, normalized to that of the replication system, can be obtained using (18) and (19). As shown in Figure 4(a), the saturation throughput of a distributed RAID system is half of that of the replication system, regardless of the array size. The saturation throughputs of the replication and RAID systems enhanced by the IDR scheme are 92% and 46% of the saturation throughput of the plain replication system, respectively. The corresponding storage-requirement factors of the different approaches, relative to the plain replication approach, are obtained by (5), (6), (15), and (16), and are shown in Figure 4(b). Note that the measures shown in Figure 4 are independent of P_f and the unrecoverable sector error probability. In contrast, the system reliability does depend on P_f and P_s. Figure 5 shows the system MTTDL, normalized to the MTTDL corresponding to the plain mirroring system, for $P_s = 4.096{\times}10^{-11}$ and $P_s = 5{\times}10^{-9}$, when $P_f = 10^{-3}$. The MTTDLs are analytically obtained using Equations (37) and (45) in [3]. Note that for $P_s = 4.096{\times}10^{-11}$, a RAID-5 system enhanced by the intra-disk redundancy scheme has a reliability that is about two orders of magnitude higher than that of a plain replication scheme. For $P_s = 5{\times}10^{-9}$, the reliability is more than one order of magnitude higher, as shown in Figure 5(b). Note also that in this case even a plain RAID-5 system offers a better reliability than a plain replication system. This seems to be counter intuitive, given that the RAID array reliability decreases as the array size increases. This behavior is explained from the fact that the system reliability depends also on the number of array groups in the system, which is smaller than the number of replication groups.

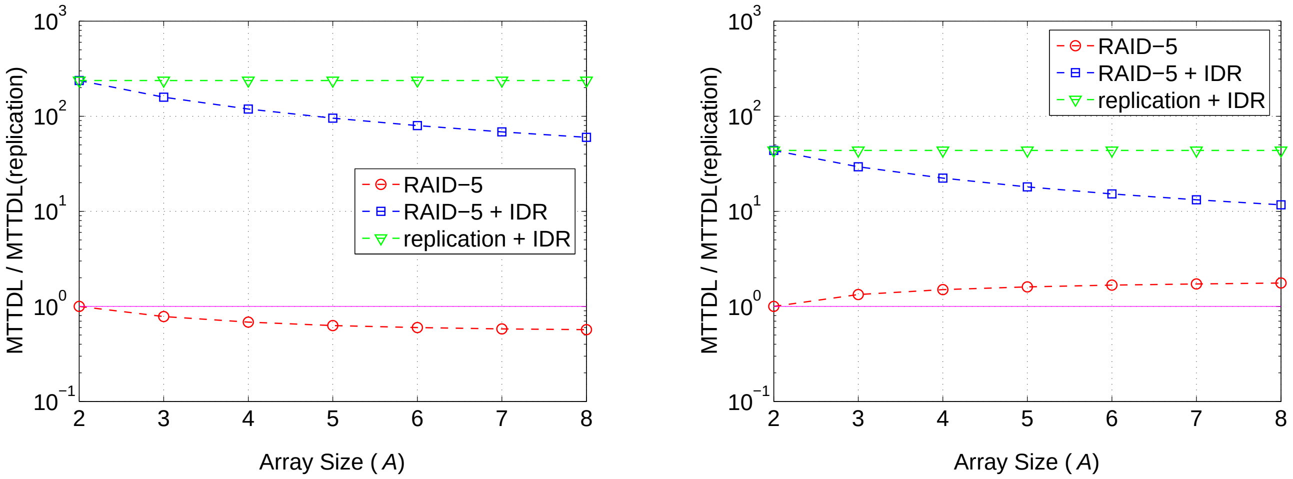
The redundancy scheme should be chosen such that sufficient degrees of storage efficiency, performance, and reliability are ensured. In general, increasing the array size in a RAID-5 system results in reduced storage space and reliability, except for large values of the unrecoverable sector error probability, in which case the reliability of a plain RAID-5 system increases, as shown in Figure 5(b). Regarding the I/O response time and throughput performance, and given



(a) Normalized saturation throughput for small-write requests.

(b) Normalized storage space required.

Figure 4: Replication and RAID-5 systems without and with IDR ($\ell = 128$, $m = 8$).



(a) $P_s = 4.096\times10^{-11}$.

(b) $P_s = 5\times10^{-9}$.

Figure 5: MTTDL vs. array size for replication and RAID-5 systems without and with IDR ($\ell = 128$, $m = 8$) for $P_f = 10^{-3}$.

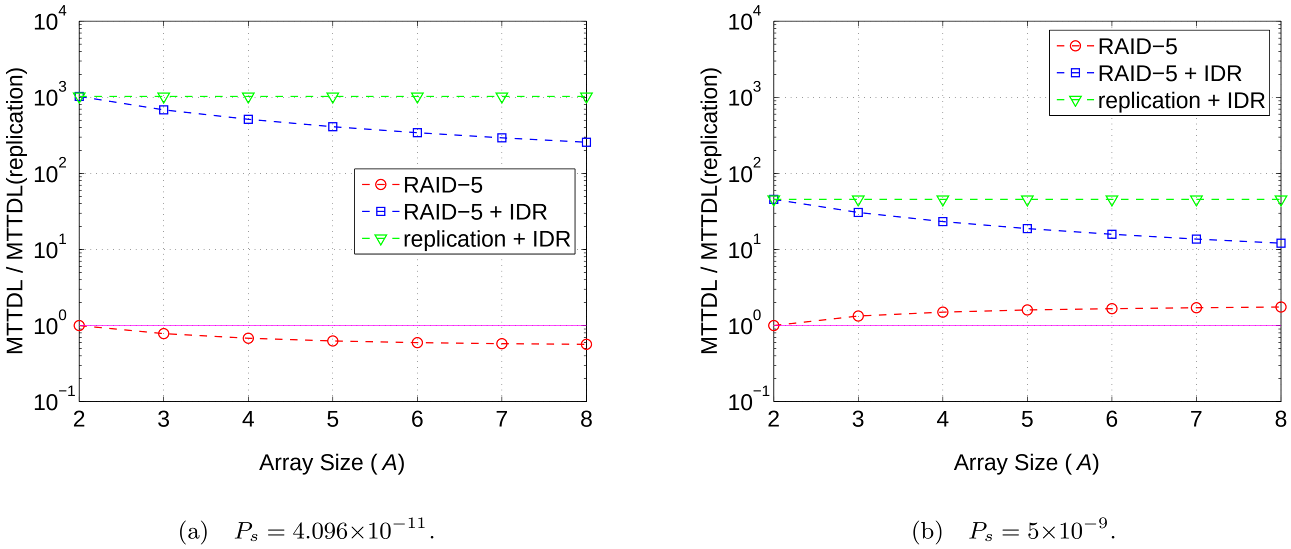
that these measures depend on the saturation throughput, we deduce from Figure 4(a) that a replication system has a better performance than a RAID system. Consequently, replacing a replication system with a RAID-5 system enhanced by the intra-disk redundancy scheme results in increased reliability and storage efficiency, but also in an increased penalty on the I/O performance. Therefore, a judicious tradeoff between these competing requirements needs to be made.

Furthermore, we have found that the same observations and conclusions hold when P_f varies. This is illustrated in Figures 6 and 7, which correspond to $P_f = 10^{-4}$ and $P_f = 10^{-2}$, respectively.

9. CONCLUSIONS

High-availability and increased reliability of today's large scale distributed storage systems require that huge amounts of user data be stored in a large number of nodes and disks in a redundant manner. Two redundancy schemes, the replication and distributed RAID schemes, were considered. Closed-form expressions for the reliability and storage efficiency of the two schemes were derived. The I/O performance was also considered by deriving analytical results for the saturation throughput of the mirroring replication and distributed RAID-5 systems. Furthermore, we also addressed the issue of placement of the redundant data in the system nodes, and demonstrated that the reliability is insensitive to the way redundant data are placed in the nodes.

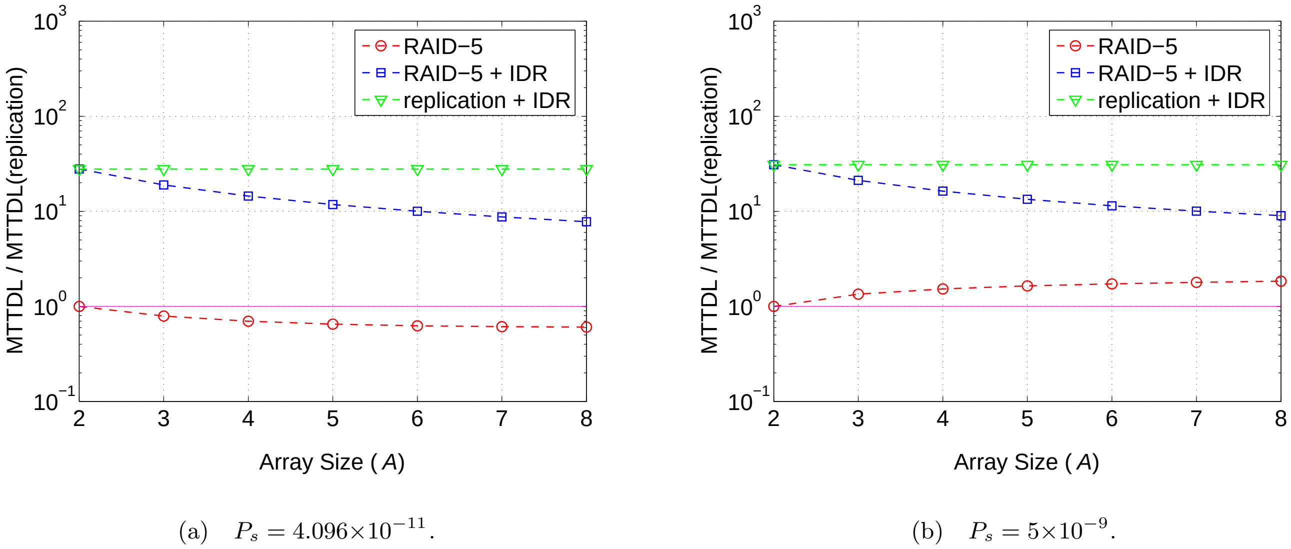
We investigated the effect of unrecoverable or latent media errors, and demonstrated that they significantly deteriorate system reliability. We therefore proposed to use the intradisk redundancy scheme, which adds another level of



(a) $P_s = 4.096{\times}10^{-11}$.

(b) $P_s = 5{\times}10^{-9}$.

Figure 6: MTTDL vs. array size for replication and RAID-5 systems without and with IDR ($\ell = 128$, $m = 8$) for $P_f = 10^{-4}$.



(a) $P_s = 4.096{\times}10^{-11}$.

(b) $P_s = 5{\times}10^{-9}$.

Figure 7: MTTDL vs. array size for replication and RAID-5 systems without and with IDR ($\ell = 128$, $m = 8$) for $P_f = 10^{-2}$.

redundancy within the disks and nodes. Our analytical results showed that distributed RAID-5 systems enhanced by the intradisk redundancy scheme provide improved reliability compared with mirroring replication systems. They also require less storage space, but incur I/O performance degradation.

10. REFERENCES

[1] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler. An analysis of latent sector errors in disk drives. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):289–300, Jun. 2007. (*Proc. ACM SIGMETRICS 2007*, San Diego, CA).

[2] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and KK Rao. Analysis of a new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors. *ACM SIGMETRICS Performance Evaluation Review*, 34(1):373–374, Jun. 2006. (*Proc. ACM SIGMETRICS 2006/Performance 2006*, Saint Malo, France).

[3] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and KK Rao. A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors. *ACM Trans. Storage*, 4(1):1–42, 2008.

[4] J. L. Hafner, V. Deenadhayalan, T. Kanungo, and KK Rao. Performance metrics for erasure codes in storage systems. IBM Res. Rep. RJ 10321, Aug. 2004.

[5] Hitachi Global Storage Technologies, Hitachi Disk Drive Product Datasheets.

http://www.hitachigst.com/. 2007.

[6] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou. Disk scrubbing versus intra-disk redundancy for high-reliability RAID storage systems. *ACM SIGMETRICS Performance Evaluation Review*, 36(1):241–252, Jun. 2008. (*Proc. ACM SIGMETRICS 2008*, Annapolis, MD).

[7] I. Iliadis and X.-Y. Hu. Reliability assurance of RAID storage systems for a wide range of latent sector errors. In *Proceedings of the International Conference on Networking, Architecture, and Storage (NAS)* (Chongqing, China), pages 10–19, Jun. 2008.

[8] M. Leslie, J. Davies, and T. Huffman. A comparison of replication strategies for reliable decentralised storage. *Journal of Networks*, 1(6):36–44, Nov./Dec. 2006.

[9] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)* (San Jose, CA), pages 17–28, Feb. 2007.

[10] S. Ramabhadran and J. Pasquale. Analysis of long-running replicated systems. In *Proceedings of the IEEE INFOCOM 2006* (Barcelona, Spain), pages 1–9, Apr. 2006.

[11] R. Rodrigues and B. Liskov. High availability in DHTs: Erasure coding vs. replication. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS)* (Ithaca, NY), pages 226–239, Feb. 2005.

[12] D. C. Sawyer. Dependability analysis of parallel systems using a simulation-based approach. *NASA-CR-195762*, Feb. 1994.

[13] B. Schroeder and G. A. Gibson. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)* (San Jose, CA), pages 1–16, Feb. 2007.

[14] T. J. E. Schwarz, Q. Xin, E. L. Miller, D. D. E. Long, A. Hospodor, and S. Ng. Disk scrubbing in large archival storage systems. In *Proceedings of the 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS)* (Volendam, The Netherlands), pages 409–418, Oct. 2004.

[15] H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)* (Cambridge, MA), pages 328–338, Mar. 2002.

[16] H. Xia and A. A. Chien. RobuSTore: A distributed storage architecture with robust and high performance. In *Proceedings of the ACM/IEEE International Conference on Supercomputing (SC),* (Reno, NV), number 44, pages 1–11, Nov. 2007.