

Disk Scrubbing Versus Intra-Disk Redundancy for High-Reliability RAID Storage Systems

Ilias Iliadis, Robert Haas, Xiao-Yu Hu, and Evangelos Eleftheriou
IBM Zurich Research Laboratory
8803 Rüschlikon, Switzerland
{ili,rha,xhu,ele}@zurich.ibm.com

ABSTRACT

Two schemes proposed to cope with unrecoverable or latent media errors and enhance the reliability of RAID systems are examined. The first scheme is the established, widely used disk scrubbing scheme, which operates by periodically accessing disk drives to detect media-related unrecoverable errors. These errors are subsequently corrected by rebuilding the sectors affected. The second scheme is the recently proposed intradisk redundancy scheme which uses a further level of redundancy inside each disk, in addition to the RAID redundancy across multiple disks. Analytic results are obtained assuming Poisson arrivals of random I/O requests. Our results demonstrate 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 a simple IPC-based intra-disk redundancy scheme, which is insensitive to the workload. In fact, the IPC-based intra-disk redundancy scheme achieves essentially the same reliability as that of a system operating without unrecoverable sector errors. For heavy workloads, the reliability achieved by the scrubbing scheme can be orders of magnitude less than that of the intra-disk redundancy scheme.

Categories and Subject Descriptors

B.3.2 [Memory Structures]: Design Styles—*mass storage* (e.g., *magnetic, optical, RAID*); B.8.1 [Performance and Reliability]: Reliability, Testing, and Fault-Tolerance; C.4 [Computer Systems Organization]: Performance of Systems—*Fault tolerance; Modeling techniques*

General Terms

Performance, Reliability

Keywords

Unrecoverable or latent sector errors, RAID, reliability analysis, MTTF, stochastic modeling

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMETRICS'08, June 2–6, 2008, Annapolis, Maryland, USA.
Copyright 2008 ACM 978-1-60558-005-0/08/06 ...\$5.00.

1. INTRODUCTION

Virtually all enterprise storage systems today make use of a RAID (redundant array of independent disks) scheme [4, 12] to protect against failures of hard-disk drives (HDDs, or disks). By far the most popular RAID scheme is currently RAID 5, which tolerates one disk failure per group (or array) of disks. Moreover, striping of user data and rotation of parity data across the array result in faster parallel accesses and better load balancing among the disks, respectively.

Total storage-system capacities tend to increase faster than disk storage-densities, hence the total number of disks rises. Moreover, disk reliability is not being improved accordingly. Therefore, disk failures inevitably increase with total storage capacity. The following trends in designing storage systems tend to further aggravate the exposure to disk failures: First, arrays grow in number of disks as a means to improve storage efficiency. Hence the exposure is aggravated owing to a higher risk of more simultaneous disk failures per array. Second, smaller disks (in terms of capacity and form factor) are used in the enterprise as a means to improve overall performance. There is therefore a higher risk from the increased total number of disks. Third, lower-cost components are being adopted in the enterprise, most notably SATA drives instead of FC and SCSI drives. SATA drives offer higher capacity per drive, but have a comparatively lower reliability [9, 13].

Constant improvements to protect against disk failures are therefore necessary. For instance, the RAID 6 scheme allows up to two disks to fail simultaneously in an array by storing two parity strips (stripe units) per stripe set [3, 5]. However, this increase in reliability reduces the storage efficiency (compared with a RAID 5 array of the same size) as well as the overall throughput performance, as each write request also requires updating the two corresponding parity units on different disks.

More precisely, failures are of the following two types: disk failures necessitating a disk replacement, where all stored data is considered lost, and errors in individual disk sectors that cannot be recovered with a re-read or the sector-based error-correction code (ECC). The percentage of drives that develop such unrecoverable or latent sector errors increases with disk capacity [13], which is particularly problematic when combined with disk failures. For example, if a disk fails in a RAID 5 array, the rebuild process must read all the data on the remaining disks to rebuild the lost data on a spare disk. During rebuild, an unrecoverable sector error on any of the remaining disks would lead to data loss. The same problem occurs when two disks fail in a RAID 6

scheme. Typical data storage installations also include a tape-based back-up or a disk-based mirrored copy at a remote location. These mechanisms can be used to reconstruct data lost because of unrecoverable errors. However, there is a significant penalty in terms of latency and throughput.

Techniques such as disk scrubbing [16, 18] and intradisk redundancy [6, 7] have been proposed to enhance the reliability of RAID systems. 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 correct them. Lost data are recovered using the RAID capability, and are subsequently written to a good disk location using the bad block relocation mechanism. Thus, the scrubbing effectively reduces the probability of encountering unrecoverable sector errors. On the other hand, the recently proposed, intra-disk redundancy scheme uses a further level of redundancy inside each disk, in addition to the RAID redundancy across multiple disks. It is based on an interleaved parity-check coding scheme [7], 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. This method introduces an additional “dimension” of redundancy inside each disk that is orthogonal to the usual RAID dimension based on redundancy across multiple disks. The RAID redundancy provides protection against disk failures, whereas the intra-disk redundancy aims to protect against media-related unrecoverable errors. Note that each of these two schemes can also be applied in conjunction with any other mechanism developed to reduce the number of unrecoverable errors and thereby improve reliability. This implies that the two schemes can also be used simultaneously.

Note that a disk drive only detects sector errors after reading them, which implies that such errors might never be detected if the corresponding sectors are not read. This is addressed by disk scrubbing, which reduces unrecoverable sector errors and works as follows: when a disk has not been accessed for some period of time, a low-priority scrubbing process is activated, which involves reading data from the disk solely for the purpose of detecting corrupted sectors. Therefore, the disk scrubbing process increases the frequency with which data are read, and this provides earlier error detection than would otherwise occur based on normal operation. Unrecoverable sector errors are now detected by both the normal read operations and the scrubbing read operations. Detecting corrupted sectors at an early stage is essential because they can be recovered using the RAID capability. If, however, they are detected during disk rebuild and while the system is in critical mode of operation, this would lead to data loss. In contrast, the intra-disk redundancy scheme in this scenario has the capability of recovering corrupted sectors, based on the intradisk redundant parity sectors, and therefore prevent data loss. The intra-disk redundancy scheme, however, does not provide an early error detection and therefore the unrecoverable sector error probability would be higher than that when scrubbing is used. Furthermore, unlike scrubbing, it requires additional capacity for storing the same amount of user data.

This paper addresses the following practical questions. What are the reliability improvements achieved by the two schemes considered? How close are the reliability levels

achieved to the absolute maximum one, which corresponds to a system operating in the absence of unrecoverable sector errors? What is the associated penalty on the I/O performance due to the additional load generated by the schemes? What is the capacity overhead of the intradisk redundancy scheme due to the additional parity sectors used?

The key contributions of this paper are the following. A new model is developed to evaluate the effectiveness of disk scrubbing, i.e. the extent to which disk scrubbing reduces the unrecoverable sector errors. The model developed captures the effect of *spatial locality*. For example, off-track writes may result in a burst of hard sector failures on the same track or cylinder of a disk. The probability of encountering unrecoverable sector errors is obtained analytically assuming Poisson arrivals of random I/O requests, which approximates the processing of disk requests from a large number of sources. It is subsequently used in conjunction with the model developed in [7] to derive the mean time to data loss (MTTDL) of RAID 5 and RAID 6 systems in the presence of unrecoverable errors and disk failures. An assessment of the reliability improvement due to scrubbing is subsequently conducted based on the MTTDL measure. Furthermore, the effectiveness of the scrubbing scheme is compared with that of the intradisk redundancy scheme. Finally, the I/O and the throughput performance of these schemes are evaluated by means of event-driven simulations under a variety of workloads.

It has been noted that substantial academic and corporate research is based on results obtained by approximate models rather than empirical data [17]. For this reason, we also studied how the theoretical results obtained compare with the empirical field results recently reported in [1, 13, 14, 17]. We have found that, for all measures considered, the theoretical results obtained by the model proposed here are in agreement with the empirical ones. This establishes a confidence for the model presented, the results obtained, and the conclusions drawn. As our results demonstrate, 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 IPC-based intra-disk redundancy scheme, which is insensitive to the workload. In fact, the IPC-based intra-disk redundancy scheme achieves essentially the same reliability as that of a system operating in the absence of unrecoverable sector errors. For heavy workloads, the reliability achieved by the scrubbing scheme can be orders of magnitude less than that of the intra-disk redundancy scheme. Moreover, the increased load of the intra-disk redundancy scheme, attributed to the longer writes due to the extra parity updates, only slightly affects the reliability and I/O performance of the system.

The remainder of the paper is organized as follows. Section 2 provides a survey of the relevant literature on scrubbing and intra-disk redundancy schemes. Section 3 describes the nature of unrecoverable errors. 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 5. The relevant performance measures are considered in Section 6. The effect of disk scrubbing on the frequency of unrecoverable sector errors is evaluated analytically in Section 7. Closed-form expressions for the probability of encountering unrecoverable sector errors are derived, and numerical results demonstrating

the effectiveness of the scrubbing scheme in reducing this probability are presented. Section 8 presents numerical results demonstrating the effectiveness of the scrubbing and intra-disk redundancy schemes in improving the reliability of the system. The I/O response time and throughput performance are evaluated by means of simulation in Section 9. Finally, we provide some relevant remarks in Section 10, and we conclude in Section 11.

2. RELATED WORK

In hard-disk drives the errors that are attributed to the read-back electronics and the magnetic medium originate primarily from the electronics noise of the front-end circuitry of the read channel, the non-ideal characteristics of the magnetic transitions (also known in the industry as media noise), thermal asperities (particle contamination and disk asperities), and medium defects [11, 8]. Some of these errors are recoverable through the error recovery procedure of a hard-disk drive (re-read process, etc.), and some are unrecoverable (hard errors). All these errors we classify as media errors to differentiate them from other sources of errors, such as firmware errors. Clearly, contamination is a prevalent cause of failures in disk drives, but its extent and the sources of errors related to it are highly dependent on the manufacturer and family of disk drives [19]. In this study, we consider contamination-related mechanisms, and in particular, thermal asperities due to contact of the head with the disk surface or particle contaminants. We also consider the error mechanisms related to transition noise (media noise), “high-fly” writes, and off-track writes, which are of more generic nature and not specific to the manufacturer and HDD family. As these factors are related to the read/write operations, the extent of the resulting errors depends on the amount of data read and written. In contrast, the model presented in [18] for assessing the disk scrubbing effect considers that sectors become erroneous according to a given fixed rate that is independent of the amount of data read/written and of the hard sector error probability. This implies that if scrubbing is not used, the probability of encountering an unrecoverable error increases in time and approaches one. In practice, however, this probability remains very small. Our aim therefore is to develop a more accurate model according to which the write operations and the hard sector error probability influence the extent to which sectors become erroneous.

An evaluation of the impact of disk scrubbing on the mean time to data loss (MTTDL) of a generic system has been presented in [2], where the term “auditing” was used instead. As the emphasis of this work was on assessing the various trade-offs rather than obtaining accurate expressions, a simple, abstract model was developed. This model accounts for *spatial overlap* of damaged areas, but it does not account for *spatial locality* of errors such as correlated media errors on the same track or cylinder of a disk [1]. Consequently, the expression derived for the MTTDL in the case of latent errors with auditing is a coarse approximation. It, in fact, underestimates the actual MTTDL because for a large scrubbing period, this expression reduces to that obtained for latent errors without auditing, which, as the authors state, underestimates MTTDL. Furthermore, these expressions refer to double-fault failure rates, which implies that they could be used to assess the MTTDL of a RAID 5 system, but not that of a RAID 6 system. Analytic expressions for the MTTDL of RAID 5 and RAID 6 systems in the pres-

ence of unrecoverable errors and disk failures were derived in [7]. These expressions, which also accounted for the effect of multiple correlated media errors, were used to obtain the MTTDL of these RAID systems when the intra-disk redundancy scheme is used. Here, we obtain the MTTDL of RAID 5 and RAID 6 systems when scrubbing is used by making use of those expressions as follows. The disk scrubbing effectively reduces the length of the time that errors remain latent, and therefore reduces the probability of encountering unrecoverable sector errors. The higher the scrubbing frequency, the higher the reduction. First, we analytically evaluate the extent of this reduction by deriving the adjusted probability of encountering an unrecoverable error. We subsequently use this probability in conjunction with the expressions mentioned above to obtain the MTTDL of RAID 5 and RAID 6 systems in the presence of unrecoverable errors and disk failures when scrubbing is used. Note also that in RAID systems, data loss occurs either when the system is in the critical mode of operation and the rebuild process tries to read a corrupted sector, or when a read request tries to read a corrupted sector that is spatially correlated with another corrupted sector or sectors. Typically, the latter event is neglected because the probability of its occurrence is orders of magnitude less than that of the former event.

3. UNRECOVERABLE ERRORS

The key problem with SATA drives is that unrecoverable errors are ten times more likely than on SCSI/FC drives [9]. 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 4.096×10^{-11} in the case of SATA drives. The above information can be interpreted as follows. If on a SATA disk 10^{14} bits, or equivalently 2.44×10^{10} sectors, are written, one of these bits is expected to suffer an unrecoverable bit error, which implies that one of these sectors is expected to suffer an unrecoverable sector error. Therefore, the probability that a bit-write operation results in an unrecoverable bit error is equal to P_{bit} . Similarly, the probability P_w that a sector-write operation results in an unrecoverable sector error is equal to P_{sec} , i.e.,

$$P_w = P_{\text{sec}} . \quad (1)$$

This probability, however, accounts only for errors occurring when data is written; transition noise (media noise), a “high-fly”, or an off-track write operation could result in a cluster of unrecoverable sector errors on a track of a disk [8]. It does not account for errors due to contamination that occur when data is read. In particular, sectors could become unreadable during read operations owing to thermal asperities, caused by contact between head and disk surface or particle contaminants [19].

There is an important distinction, however, between the types of errors described above. Unrecoverable sector errors created by write operations remain latent, whereas unrecoverable sector errors that occur when data is read are recovered and subsequently written to good disk locations using the bad block relocation mechanism. In the remainder, we therefore consider that a latent unrecoverable error on a given sector at an arbitrary time is related to the write process.

4. SCRUBBING SCHEME

Here we briefly review the scrubbing scheme. The scrubbing process identifies unrecoverable sector errors at an early stage and attempts to correct them. The scrubbing process is periodic in that sectors are scrubbed at fixed intervals. Two different disk scrubbing schemes are considered: a *sequential* (or *deterministic*) and a *random* one.

According to the sequential scheme, the scrubbing process reads and checks sectors in order, starting from the first and then ending at the last. The process then starts over by checking the first sector. The rate at which sectors are checked is specified a priori. This rate and the disk capacity determine the *scrubbing period*, i.e. the time required for a complete check of all sectors of a disk. Note that any arbitrarily given sector is checked regularly, with the intervals between successive checks being fixed and equal to the scrubbing period. Note that this also holds in the case of a deterministic, not necessarily sequential, scrubbing scheme that ensures that the whole disk is scrubbed within the scrubbing period. For this reason, we refer to the deterministic scheme in the remainder of the paper. The frequency at which any arbitrarily given sector is checked is inversely proportional to the scrubbing period.

According to the random scheme, the scrubbing process checks randomly chosen sectors. Note that here, as in the deterministic scheme, the number of sectors checked in a scrubbing period is equal to the number of sectors contained in a disk. However, in contrast to the deterministic scheme, the random scheme does not guarantee that all sectors of a disk are checked in a scrubbing period. It may well be that some sectors will be checked multiple times, and other sectors not at all.

5. INTRA-DISK REDUNDANCY SCHEME

Here we briefly review the intra-disk redundancy (IDR) scheme presented in [6] 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 [7], 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. These parity sectors are written simultaneously with the corresponding data by a single I/O request, and therefore do not require extra write operations. The traditional single-parity-check (SPC) coding scheme corresponding to $m = 1$ is also considered. 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 size of a segment and the number of parity sectors in a segment are chosen to be equal to $\ell = 128$ and $m = 8$, respectively, to ensure sufficient degrees of storage efficiency, performance and reliability. The choice of $m = 8$ is rea-

sonable given that recent empirical data indicate that the median number of errors for error disks is 3 [1]. The storage efficiency $se^{(IDR)}$ of the intra-disk redundancy scheme is then given by $(\ell - m)/\ell$, which is equal to 94%.

6. SYSTEM ANALYSIS

The notation used for the purpose of our 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.

The storage efficiency of the RAID scheme chosen is given by

$$se^{(RAID)} = \frac{N - p}{N} = 1 - \frac{p}{N}, \quad (2)$$

with

$$p = \begin{cases} 1 & \text{for a RAID 5 system} \\ 2 & \text{for a RAID 6 system.} \end{cases} \quad (3)$$

Note that the above expressions hold for a scheme not using intra-disk redundancy. If an intra-disk redundancy scheme is used, the overall storage efficiency of the entire array (or system) is given by

$$se^{(RAID+IDR)} = se^{(RAID)} se^{(IDR)} = \left(1 - \frac{p}{N}\right) \left(1 - \frac{m}{\ell}\right). \quad (4)$$

The number of sectors in a disk, S_D , is given by

$$S_D = \left\lfloor \frac{C_d}{S} \right\rfloor, \quad (5)$$

the ratio of disk drive capacity to sector size. Therefore the number of data sectors, S_d , in a disk of a RAID array is given by

$$S_d = \left\lfloor se^{(RAID)} S_D \right\rfloor, \quad \text{or} \quad S_d = \left\lfloor se^{(RAID+IDR)} S_D \right\rfloor, \quad (6)$$

depending on whether intra-disk redundancy is used.

We assume that the I/O requests in a disk are random, uniformly distributed across the disk, and arrive at a rate of σ according to a Poisson process. This approximates the processing of disk requests from a large number of sources. Let R denote the request size (in number of sectors), E_r the number of unrecoverable sector errors due to contamination that occur in the case of a read request, and E_w the number of sectors that are erroneously written in the case of a write request. Note that such errors may occur in clusters. Consequently, the load h of a given data sector, or, equivalently, the rate at which a data sector is read/written, is given by

$$h = \frac{\bar{R} \sigma}{S_d}, \quad (7)$$

i.e. the ratio of the rate at which data sectors are read/written to the number of data sectors contained in a disk. In the remainder, unless otherwise indicated, the term sector will only refer to the data sectors.

Note also that the probability P_w that a sector-write operation results in an unrecoverable sector error is given by

$$P_w = \frac{\bar{E}_w}{\bar{R}}. \quad (8)$$

Table 1: Notation of system parameters

Parameter	Definition
N	Number of disks per array group
n_G	Number of array groups in the system
C_d	Disk drive capacity
S	Sector size
ℓ	Number of sectors in a segment
m	Number of parity sectors in a segment, number of interleaves, or interleaving depth
$1/\lambda$	Mean time to failure for a disk
P_{bit}	Probability of an unrecoverable bit error (data sheet specification)
P_{sec}	Probability of an unrecoverable sector error (data sheet specification)
T_s	Scrubbing period for a disk
σ	Rate of I/O requests in a disk (workload)
R	The size of an I/O request
E_r	The number of unrecoverable sector errors due to contamination that occur by a read request
E_w	The number of unrecoverable sector errors that occur by a write request
r_w	Ratio of write operations to read/write operations
h	Rate at which a disk sector is read/written (load)
$se^{(\text{RAID})}$	Storage efficiency of the RAID scheme
$se^{(\text{IDR})}$	Storage efficiency of the intra-disk redundancy scheme
$se^{(\text{RAID+IDR})}$	Overall storage efficiency of the entire system
S_D	Number of sectors in a disk
S_d	Number of data sectors in a disk
$1/\mu$	Mean time to rebuild in critical mode for a RAID 5 array
$1/\mu_1$	Mean time to rebuild in degraded mode for a RAID 6 array
$1/\mu_2$	Mean time to rebuild in critical mode for a RAID 6 array
P_w	Probability of an unrecoverable sector error due to a write operation
$P_s(t)$	Probability of an unrecoverable error on a tagged sector at time t
P_s	Probability of an unrecoverable error on a tagged sector at an arbitrary time
P_e	Probability of an unrecoverable error on a tagged sector at an arbitrary time when scrubbing is not used

7. SCRUBBING ANALYSIS

Here we consider the effect of the scrubbing process and derive the probability P_s of an unrecoverable error on a given sector at an arbitrary point in time. Assuming that sectors are read/written at random and according to a Poisson process, closed-form expressions for P_s for the two scrubbing schemes are obtained by the following propositions.

PROPOSITION 1. *For the deterministic scrubbing scheme, it holds that*

$$P_s = \left(1 - \frac{1 - e^{-hT_s}}{hT_s}\right) P_e, \quad (9)$$

where

$$P_e = r_w P_w. \quad (10)$$

PROOF. See Appendix A. \square

PROPOSITION 2. *For the random scrubbing scheme, it holds that*

$$P_s = \frac{hT_s}{1 + hT_s} P_e, \quad (11)$$

where P_e is given by (10).

PROOF. See Appendix B. \square

Remark 1. For both scrubbing schemes, P_s depends only on P_w , r_w , and the product hT_s , i.e. the ratio of the rate a sector is read/written to the rate it is scrubbed.

Remark 2. From (9) and (11), it follows that the probability for an unrecoverable sector error for the deterministic scheme is always smaller than that for the random scheme. Furthermore, for small values of the product hT_s such that $hT_s \ll 1$, these probabilities can be approximated by

$$P_s \approx \frac{1}{2} P_e hT_s \quad (12)$$

for the deterministic scrubbing scheme, and by

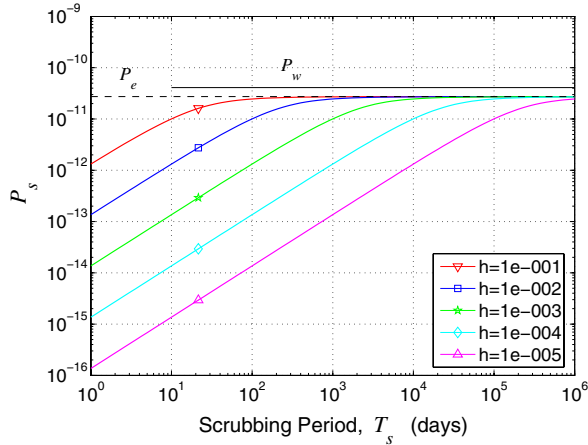
$$P_s \approx P_e hT_s \quad (13)$$

for the random scrubbing scheme.

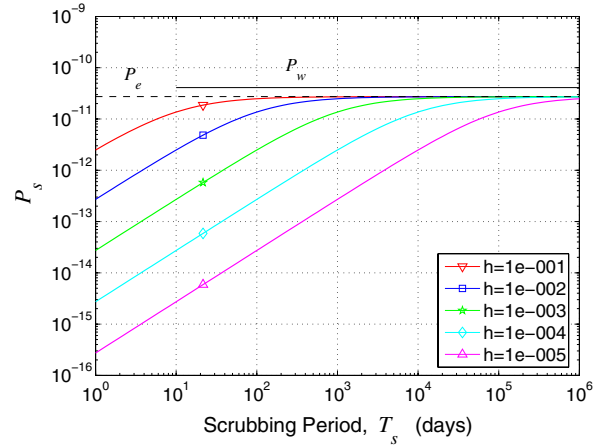
Remark 3. The probability of an unrecoverable sector error when scrubbing is not used, can be obtained by taking the limit $T_s \rightarrow \infty$ in either (9) or (11), and is equal to $P_s = P_e$. It therefore depends on the ratio of read to write operations, but not on the workload. Increasing the workload results in an increase of the read/write operations, which on the one hand increases the unrecoverable sector errors created, but on the other hand reduces the unrecoverable sector errors due to correction of existing ones.

Remark 4. From (10), it follows that P_e is bounded above by the unrecoverable sector error probability P_w , which corresponds to the case where only write operations are performed and scrubbing is not used, i.e. $P_e \leq P_w$. This, together with (9) and (11), implies that also P_s is bounded above by P_w , regardless of the ratio of read to write operations, i.e.,

$$P_s \leq P_e \leq P_w. \quad (14)$$



(a) Deterministic scrubbing.



(b) Random scrubbing.

Figure 1: P_s as a function of T_s for $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} (read/write operations per sector per day), and $r_w = 0.66$.

7.1 Numerical Results

We consider SATA drives with $C_d = 300$ GB, $S = 512$ B, and data sheet specification $P_{\text{bit}} = 10^{-14}$ and $P_{\text{sec}} = 4.096 \times 10^{-11}$. Based on this, and by making use of (5), the probability that a disk contains at least one hard sector error is approximately equal to the product $S_D P_{\text{sec}}$, which is equal to 2.4%. This result is in agreement with the empirical data reported in [13, 1] where it is found that about 2% and 3.45% of the disks, respectively, ever developed a latent sector error. From (1), it follows that the probability P_w that a sector-write operation results in an unrecoverable sector error is equal to 4.096×10^{-11} .

Figure 1 shows the unrecoverable sector error probability P_s at an arbitrary time as a function of the scrubbing period T_s for the deterministic and random scrubbing schemes, as derived from (9) and (11). In the remainder, it is assumed that the ratio of read to write operations is set to be 1:2, i.e. there are 33.33% reads and 66.67% writes, which yields $r_w = 0.66$. Empirical data reported in [14] indicate a large variation of this ratio. We consider a smaller number of read operations than write operations because a front-end cache reduces the number of read requests sent to the disks. The horizontal solid line in the figures indicates P_w , the SATA drive specification for unrecoverable sector errors. It can be seen that, regardless of the load h (measured in days), the use of scrubbing results in reduced unrecoverable sector error probabilities. Clearly, the smaller the scrubbing period, the lower the unrecoverable sector error probability. However, as T_s increases, P_s increases and approaches, according to (9) and (11), P_e , the unrecoverable sector error probability when disk scrubbing is not used. This value is indicated by the horizontal dashed line and, according to (14), is always less than or equal to P_w ; it approaches P_w when r_w approaches one. As the load h increases, P_s also increases because the higher the workload of the system, the larger the number of write operations, and therefore the larger the net number of additional unrecoverable sector errors. According to Remark 2, the unrecoverable sector error probability in the case of deterministic scrubbing is always smaller than that when random scrubbing is used. In particular, for small values of T_s , it is practically half of that

Table 2: Parameter values

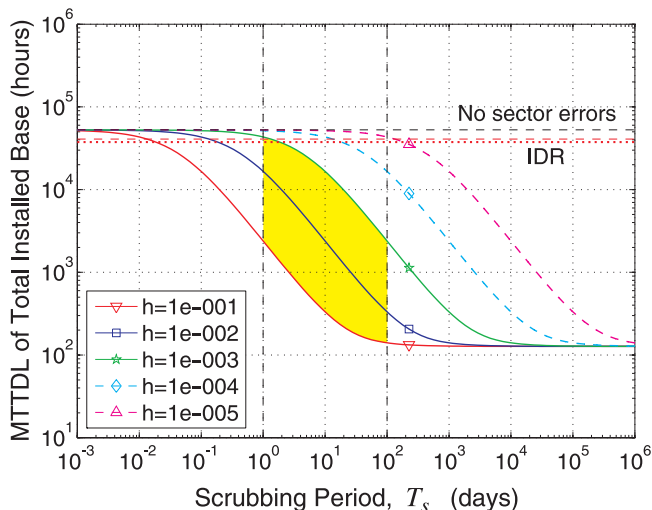
Parameter	Value
$1/\lambda$	500,000 h
C_d	300 GB
N	8 (for RAID 5), 16 (for RAID 6)
$1/\mu$	17.8 h
$1/\mu_1$	17.8 h
$1/\mu_2$	17.8 h
S	512 bytes = 4096 bits
P_w	4.096×10^{-11}
P_e	2.731×10^{-11}

when random scrubbing is used, as suggested by (13) and (12). For this reason, it is the deterministic scrubbing that is widely used in practice, and we therefore consider only this scheme in the remainder of the paper.

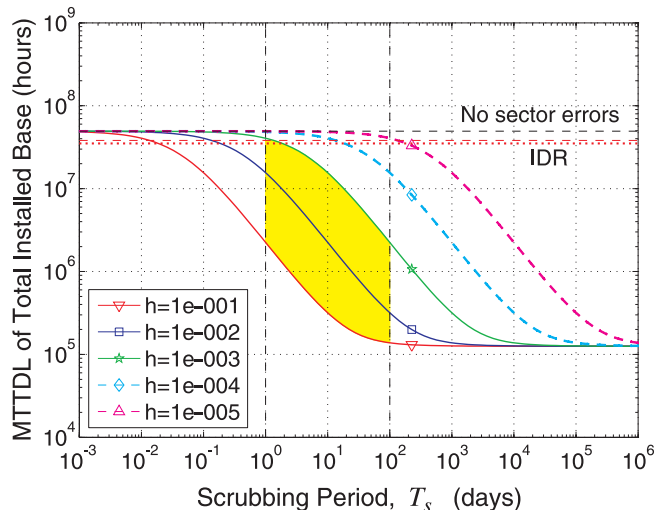
8. RELIABILITY RESULTS

Here we analytically assess the effectiveness of the disk scrubbing scheme in improving the reliability of RAID systems, and also compare it with the intradisk redundancy scheme through illustrative examples. The reliability of a RAID system is assessed in terms of the MTTF, which clearly depends on the size of the system. It turns out that the MTTF scales with the inverse of the system size. For example, increasing the system size by a given factor will result in an MTTF decrease by the same factor. Consequently, for the purpose of studying the behavior of the two schemes, the choice of the system size is not essential. Also, the conclusions drawn regarding the performance comparison are independent of the system size chosen. We proceed by considering an installed base of systems using SATA disk drives and storing 10 PB of user data. The corresponding parameter values for the SATA disks, which are summarized in Table 2, are obtained from [7].

We note that for disk drives running 24 hours a day, 7 days a week, a mean time to failure (MTTF) of 500,000 hours corresponds to an annualized failure rate (AFR) of 1.75%,



(a) RAID 5.



(b) RAID 6.

Figure 2: MTTDL as a function of T_s for deterministic scrubbing and $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} (read/write operations per sector per day), under correlated unrecoverable sector errors and $r_w = 0.66$.

which is in agreement with the AFR of 1.7% observed for drives that were in their first year of operation [13]. Furthermore, the data collected reveal that the failure rate does not necessarily increase as disk utilization increases. In particular, after the first year, the AFR of high-utilization drives is at most moderately higher than that of low-utilization drives. Consequently, variation of the scrubbing period and the workload is not expected to influence the MTTF of the disks.

The effect of the spatial locality of correlated media errors is also considered. Adopting the notation used in [7], let $\{b_j\}$ denote the probability density function of the length j of a typical burst of consecutive errors, and \bar{B} the corresponding average length. Moreover, let G_n denote the probability that the length of a burst is greater than or equal to n , i.e. $G_n \triangleq \sum_{j=n}^{\infty} b_j$, for $n = 1, 2, \dots$. We now consider the following error-burst length distribution, based on actual data collected from the field for a product that is currently being shipped [7]:

$$\mathbf{b} = [0.9812 \ 0.016 \ 0.0013 \ 0.0003 \ 0.0003 \ 0.0002 \ 0.0001 \\ 0.0001 \ 0 \ 0.0001 \ 0 \ 0.0001 \ 0.0001 \ 0 \ 0.0001 \ 0.0001]. \quad (15)$$

Then, we have bursts of at most 16 sectors with $\bar{B} = 1.0291$, $\bar{B}^2 = 1.1771$, and $G_9 = 0.0005$. As the parameter m of the IPC-based redundancy scheme is chosen to be equal to 8, the probability that this scheme will not be able to correct a single burst of consecutive errors occurring in a segment is equal to G_9 , i.e. 0.0005. Note that the issue of spatial locality of errors has been studied in [1]. For a nearline disk family, the average number of neighboring errors within a range of 10 KB (approx. 20 sectors) of an existing sector in error is 0.17. It can be shown that, using the terminology defined, this measure is given by $\bar{B}^2/\bar{B} - 1$, which, for the error-burst length distribution given in (15), is equal to 0.14.

From (2), (3), and (4), it follows that the storage efficiency of the entire system is independent of the RAID configuration if the arrays in a RAID 6 system are twice the size of those in a RAID 5 system. For a RAID 5 system with

$N = 8$, when no intra-disk redundancy is used, the required number of arrays, n_G , to store the user data is equal to 4762 (i.e. 10 PB/(7×300 GB)), whereas for a RAID 6 system with $N = 16$, it is equal to 2381 (i.e. 10 PB/(14×300 GB)). The corresponding storage efficiency is equal to 7/8, i.e. 0.875. For the IPC intra-disk redundancy scheme, with a segment comprised of $\ell = 128$ sectors and $m = 8$ parity sectors in a segment, the intra-disk storage efficiency is equal to 0.94. Furthermore, the required number of arrays, n_G , for a RAID 5 configuration is obtained as the ratio of 4762 to the intra-disk storage efficiency and is equal to 5080. Similarly, for a RAID 6 configuration, the required number of arrays is equal to 2540. The overall storage efficiency is obtained by (4) and is equal to 0.82.

The system reliability is assessed in terms of the MTTDL, which is analytically obtained by the closed-form expressions (Equations (37), (45) and (52)) derived in [7]. The MTTDL corresponding to a RAID system operating in the absence of unrecoverable sector errors is equal to 52,696 hours in the case of the RAID 5 system and 4.9×10^7 hours in the case of the RAID 6 system. These values are indicated in Figure 2 by the upper dashed lines. For the configuration considered, and with $P_s = P_e$, it turns out that the presence of unrecoverable errors causes the MTTDL to decrease significantly, namely, by more than two orders of magnitude, from 52,696 to 127 hours in the case of the RAID 5 system and from 4.9×10^7 to 1.2×10^5 hours in the case of the RAID 6 system. Note that, according to Remark 3, this drastic MTTDL reduction is independent of the workload. Note also that the MTTDL of a RAID 6 system in the presence of unrecoverable errors is higher than that of a RAID 5 system operating in the absence of unrecoverable sector errors. However, these two MTTDL values are of the same order of magnitude. The MTTDLs corresponding to the RAID 5 and RAID 6 systems enhanced by the IPC-based intra-disk redundancy scheme are derived based on the above mentioned analytic expressions of [7], and are found to be equal to 40,760 and 3.8×10^7 hours, respectively. These values are indicated in Figure 2

by the lower dashed lines. Consequently, the IPC intra-disk redundancy scheme improves the MTTDL by more than two orders of magnitude, which practically eliminates the negative impact of unrecoverable sector errors. Note that by taking into account the longer writes due to the extra parity updates of the intra-disk redundancy scheme, according to Remarks 3 and 4, the probability of unrecoverable sector errors will increase but not exceed P_w . The MTTDL corresponding to $P_s = P_w$ is found to be equal to 37,491 hours for the RAID 5 system, and 3.5×10^7 hours for the RAID 6 system, indicated by the horizontal dotted line. Consequently, the actual MTTDL will be in the small region between the horizontal dashed and dotted lines, which, in turn, implies that the reliability level is practically insensitive to the workload.

We now explore the effectiveness of disk scrubbing in improving the reliability of RAID 5 and RAID 6 systems in the presence of unrecoverable errors and disk failures. The system reliability, in terms of the MTTDL, is analytically obtained by substituting the value for the unrecoverable sector error probability P_s given by (9) into the closed-form expressions derived in [7]. The effect of deterministic disk scrubbing on the MTTDL when intra-disk redundancy is not used can be seen in Figure 2 as a function of the scrubbing period. When the scrubbing period is extremely large, the scrubbing scheme is not effective, and the MTTDL approaches that of a system without scrubbing. It can be seen that, regardless of the workload and for both RAID 5 and RAID 6 systems, the MTTDL increases as the scrubbing period decreases. In particular, the MTTDL for small scrubbing periods is more than two orders of magnitude higher than that for large scrubbing periods. This is because frequent scrubbing results in reducing the probability of unrecoverable sector errors. Thus, as T_s decreases, P_s also decreases, and therefore the MTTDL increases, approaching the maximum possible value (indicated by the upper dashed line), which corresponds to the MTTDL of a RAID system operating in the absence of unrecoverable sector errors.

When a disk is scrubbed every day and the load h is light, not exceeding 0.001, the MTTDL is improved significantly and approaches the upper dashed line, which implies that the negative impact of the unrecoverable sector errors is practically eliminated. When the load h is 0.01, however, the MTTDL is improved by two orders of magnitude, but does not reach the level achieved by the IPC-based intra-disk redundancy scheme, indicated by the lower horizontal dashed line. Furthermore, when the load h is 0.1, the MTTDL is improved by only one order of magnitude, and is therefore significantly less than the MTTDL obtained by the intra-disk redundancy scheme. Note also that owing to the physical operational constraints of the system, and as we will see in the next section in more detail, the scrubbing period cannot be arbitrarily small. This implies that the scrubbing mechanism may not be able to reduce the number of unrecoverable errors sufficiently and therefore eliminate their negative impact on the MTTDL.

Clearly, as Figure 2 also captures areas that are not realistic, we now proceed to identify the areas of practical importance. According to [1], the scrubbing process scans the entire surface of the media at least once every two weeks. Consequently, a realistic scrubbing period should be between one and 100 days. The corresponding region for T_s is indicated in Figure 2 between the two vertical dashed lines.

A characterization of disk drive workloads in various system environments is presented in [14]. The study of traces in enterprise and consumer electronics environments reveals that the mean interarrival times of I/O requests in a disk are in the range of 56.0 to 246.6 ms. This implies that the mean arrival rate of requests per disk per second, σ , is in the range of $4.05 \leq \sigma \leq 17.57$. The load h (measured in days) is obtained as a function of σ by making use of (2), (3), (5), (6), and (7), as follows:

$$\begin{aligned} h &= \frac{s_{\text{req}} \cdot SD}{S_d S} \sigma = \frac{s_{\text{req}} \cdot SD}{se^{(\text{RAID})} C_d} \sigma \\ &= \frac{4 \text{ KB} \times (24 \times 60 \times 60)}{7/8 \times 300 \text{ GB}} \sigma \\ &= 0.0013 \sigma, \end{aligned} \tag{16}$$

where s_{req} is the size of a request, SD the number of seconds in a day, and the product in the denominator of the fraction is the disk capacity that is effectively used for user data storage.

From the above, it now follows that the load h corresponding to σ is in the range of $5.27 \times 10^{-3} \leq h \leq 2.28 \times 10^{-2}$. This implies that the region of practical importance for h is the one shown in Figure 2 between the solid curves corresponding to $h = 10^{-1}$ and 10^{-3} .

By considering the above bounds for T_s and h , and inspecting the area of practical interest, which is indicated by the shaded region, we conclude that the scrubbing mechanism does not reduce the number of unrecoverable errors sufficiently so as to eliminate their negative impact on the MTTDL. For typical workloads and scrubbing frequencies, the MTTDL can be improved significantly, but does not reach the level achieved by the IPC-based intra-disk redundancy scheme. This improved MTTDL can be smaller than the MTTDL obtained by the intra-disk redundancy scheme by two orders of magnitude.

Furthermore, we have found that the same conclusions hold when the systems with intra-disk redundancy are compared with systems employing scrubbing and having comparable storage efficiencies. A RAID 5 system with $N = 6$ and a RAID 6 system with $N = 12$ have a storage efficiency of 0.83. The MTTDLs improve, as expected, but by only a factor of approximately $8/6 = 1.33$ and $(16 \cdot 15)/(12 \cdot 11) = 1.81$, respectively.

9. PERFORMANCE RESULTS

Here we study the performance impact of the deterministic scrubbing scheme on the response time and the saturation throughput of a RAID 5 system by using event-driven simulations. We do not present results for a RAID 6 system because the conclusions drawn based on the results presented below for a RAID 5 system apply equally in the case of a RAID 6 system.

Most modern RAID controllers have a large battery-backed cache that boosts the overall system performance by reducing the I/O requests to the disks and performing aggressive read-ahead and write-behind. The response time of an array as experienced by the end user can be dramatically shortened by increasing the size of the array cache and selecting the replacement strategy based on the characteristics of workloads. As our main interest in the simulation is the difference in performance of the RAID schemes considered, rather than caching mechanism or characteristics of

workloads, we start measuring the response time of requests after caching, i.e., from the instant when they are sent to the disks. Therefore, the saturation throughput measures the maximum throughput between the front-end (cache) and the back-end (disk array), assuming sufficient bandwidth in between. The higher the saturation throughput, the better the performance of the underlying RAID mechanism. Note also that, in contrast to read operations, scrubbing operations do not require the transfer of data from disk to array controller, unless the rare case in which an error is identified.

We have developed a lightweight event-driven simulator. Various standard RAID simulators are publicly available in the community, such as the CMU’s DiskSim [20], and the HP Labs’ Pantheon [10] for disk arrays. With the advent of the C++ standard library and the concept of generic programming, particularly the standard template library (STL), developing a lightweight event-driven simulator from scratch turns out to be an easier task than understanding and tailoring an existing large software package for our purpose. We have also built an HDD module targeted for an industry brand 300 GB SATA drive, following the approach described in [15] and consulting the source code of DiskSim. The disk-drive model captures major features such as zoned cylinder allocation, mechanical positioning parameters such as seek time, settling time, cylinder and head skew, as well as rotational latency, data transfer latency, and buffering effects such as read ahead. The simulated response time of the HDD exhibits a good match with its nominal specification. We assume a first-come first-served (FCFS) scheduling policy for serving the I/O requests at each disk. Actually, we have tested several other disk-scheduling policies such as SSTF, LOOK, and C-LOOK, and have found that the scheduling policy does not change the relative performance of disk scrubbing and intra-disk redundancy schemes.

We obtain the mean response time of a RAID 5 array consisting of 8 SATA disks as a function of the scrubbing period. We also evaluate the performance of the plain RAID 5 scheme as well as of the RAID 5 scheme enhanced by the addition of the intra-disk redundancy scheme. For the intra-disk redundancy scheme, we employ an IPC intra-disk redundancy scheme with a segment size of 128 sectors, comprising 8 redundant sectors and 120 data sectors. We consider the small-write scenario and use synthetic workloads generating aligned 4-KB-small I/O requests with uniformly distributed logical block addresses (LBAs). The ratio of read to write is set to be 1:2, i.e. there are 33.33% reads and 66.67% writes. The request inter-arrival times are assumed to be exponentially distributed.

Figure 3 shows the average response time of a RAID 5 system enhanced either by a scrubbing or an IPC-based intra-disk redundancy scheme as a function of σ , which is measured in I/O requests per disk per second. The increase observed in the response times when RAID 5 is enhanced by the IPC intra-disk redundancy scheme is minor for mean arrival rates of less than 25 I/O requests per second. Also the saturation throughput for RAID 5 is 30.25 I/O requests per disk per second, whereas for RAID 5 enhanced by the IPC intra-disk redundancy scheme it is 29.75 I/O requests per disk per second, as indicated by the horizontal dashed lines in Figure 4. This represents a minor, 2% degradation in saturation throughput due to the IPC intra-disk redundancy scheme and is incurred by the longer writes because of the extra parity updates required. Similarly, a minor degra-

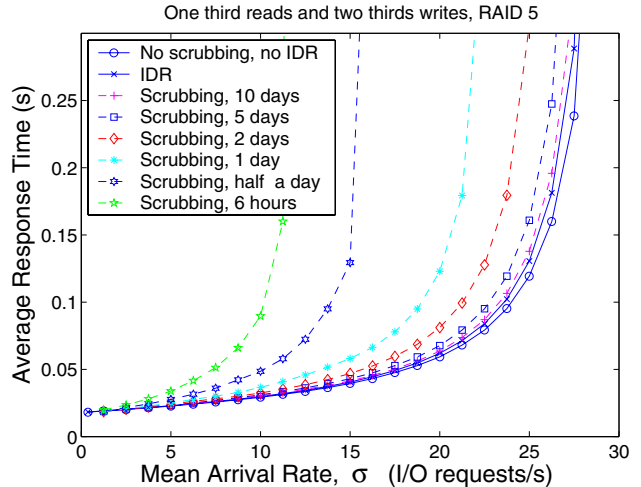


Figure 3: Response time of a RAID system for various scrubbing periods (synthetic workload, small writes).

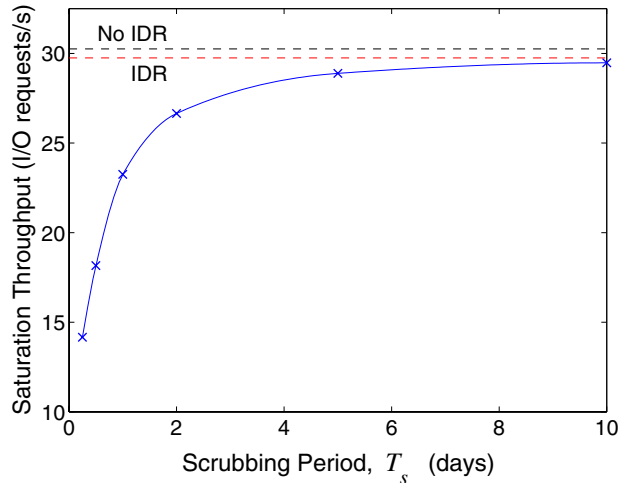


Figure 4: Saturation throughput as a function of the scrubbing period.

ation in saturation throughput is observed when the scrubbing scheme is used with a large scrubbing period. For large scrubbing periods, the additional workload due to scrubbing is negligible and therefore the mean response times are practically unaffected. For example, for a scrubbing period of 10 days ($T_s = 10$), Figure 3 shows that the mean response times are slightly higher than the ones obtained when scrubbing is not used. As the scrubbing period T_s decreases, the corresponding workload increases and, therefore, the mean response time also increases. Furthermore, as T_s decreases, the saturation throughput decreases, as shown in Figure 4. For example, for a scrubbing period of one day, the saturation throughput is 23.25 I/O requests per disk per second, which is significantly less (78%) than that of the IPC intra-disk redundancy scheme. For a scrubbing period of half a day, the saturation throughput is 18.12 I/O requests per

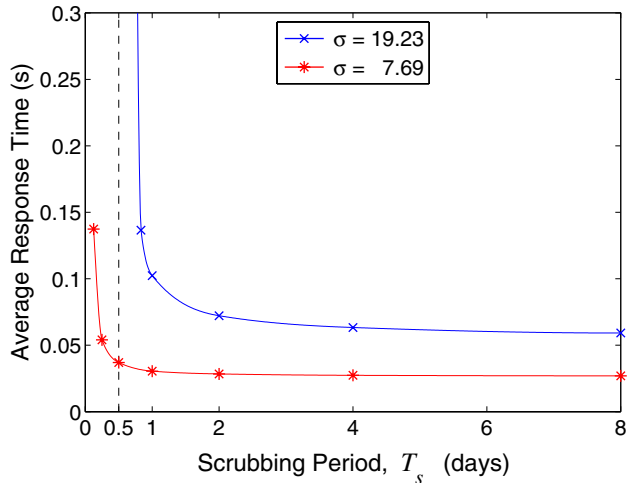


Figure 5: Response time of a RAID 5 system as a function of the scrubbing period (synthetic workload, small writes, one third reads, two thirds writes).

disk per second, which is almost only half of that of the IPC intra-disk redundancy scheme. This implies that, for a given workload, the scrubbing period cannot be arbitrarily small. For example, for a workload of 19.23 I/O requests per second the scrubbing period can be one day but not half a day.

The fact that for a given workload the scrubbing period cannot be arbitrarily small is also demonstrated in Figure 3. For example, for a workload of 19.23 I/O requests per second, which corresponds to a load $h = 0.025$, Figure 3 shows again that the scrubbing period can be as low as one day, but not half a day because as the total workload exceeds the system capacity, the corresponding response time grows to infinity. Although, for a smaller workload of 7.69 I/O requests per second, which corresponds to a load $h = 0.010$, the scrubbing period can be as low as half a day or even a quarter of a day. Consequently, the scrubbing period is lower bounded, with the bound depending on the workload of the system. Note that for a workload of 19.23 I/O requests per second ($h = 0.025$) and for a scrubbing period of one day ($T_s = 1$), the average response time increases from 0.059 to 0.103 seconds, as shown in Figure 5, but also the reliability improves with the MTTDL increasing from 127 to 7780 hours, indicated by the square symbol in Figure 6. The resulting MTTDL is, however, still an order of magnitude smaller than the 52,696 hours corresponding to the MTTDL of the system operating in the absence of unrecoverable sector errors. Similarly, for a smaller workload of 7.69 I/O requests per second ($h = 0.010$) and for a scrubbing period of half a day ($T_s = 0.5$), the average response time increases from 0.027 to 0.037 seconds, but also the reliability improves, with the MTTDL increasing from 127 to 24,855 hours, indicated by the triangle symbol in Figure 6. But even in this case, the resulting MTTDL is considerably lower than the target of 52,696 hours.

Note also that increasing the scrubbing period, and considering a realistic scrubbing period of one week, it practi-

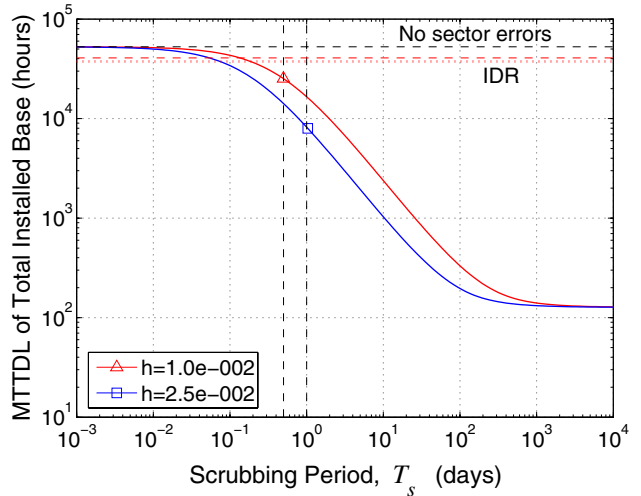


Figure 6: MTTDL as a function of T_s for deterministic scrubbing and $h = 0.010$ and 0.025 , under correlated unrecoverable sector errors and $r_w = 0.66$.

cally eliminates the increase in the average response time, but it causes the MTTDL to decrease significantly, namely, to 1430 and 3320 hours for the workloads of 19.23 and 7.69 I/O requests per second, respectively.

From the above, it follows that the scrubbing period should be chosen such that sufficient degrees of performance and reliability are ensured. First, reducing the scrubbing period 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. Second, the degrees of performance and reliability provided are reduced as the workload of the system increases. Consequently, under heavy workload conditions, the scrubbing mechanism will not be able to provide the desired level of reliability.

10. DISCUSSION

The current technological trend in hard-disk drives exhibits strong evidence that the capacity is growing at a fast pace, whereas the I/O performance improvement is moderate. Ultimately, this would result in imposing more stringent constraints on the scrubbing frequency, which in turn would have a negative impact on the performance of future storage systems. This trend clearly advocates the use of intra-disk redundancy to cope with media-related unrecoverable errors. On the other hand, in storage systems operating under light loads, scrubbing can be employed to achieve a reliability level which is close to the absolute maximum one, even closer than that of the intra-disk redundancy.

Note that both, the scrubbing scheme and the intra-disk redundancy scheme, can also be applied in conjunction with any other mechanism developed to reduce the number of unrecoverable errors and thereby improve reliability. This implies that the two schemes can also be used simultaneously. For example, in the case where the scrubbing scheme alone cannot reach the desired level of reliability, introducing in addition the intra-disk redundancy scheme could serve the purpose. For the case studied in this paper, however, the

results obtained do not seem to justify the need for the converse, as the intra-disk redundancy scheme alone already provides a degree of reliability close to the optimal, with minimal impact on the I/O system performance.

11. CONCLUSIONS

Today's data storage systems are increasingly adopting low-cost disk drives that have higher capacity but lower reliability, leading to more frequent rebuilds and to a higher risk of unrecoverable media errors. The disk scrubbing and intradisk redundancy schemes, which were developed to enhance the reliability of RAID systems, were considered. A new model capturing the relation between the write operations and the appearance of hard sector errors was introduced. The effect of disk scrubbing on reducing the frequency of unrecoverable sector errors was assessed analytically under the assumption of Poisson arrivals of random I/O requests. Closed-form expressions for the probability of encountering unrecoverable sector errors were derived for the deterministic and random scrubbing schemes. The effectiveness of the scrubbing scheme in improving the reliability, in terms of the mean time to data loss, of RAID 5 and RAID 6 systems in the presence of unrecoverable errors and disk failures was explored, and also compared with that of the intradisk redundancy scheme.

Our results demonstrate that the reliability improvement due to disk scrubbing depends on the scrubbing frequency and the workload of the system. In particular, for typical scrubbing frequencies and workloads, the reliability improvement due to disk scrubbing does not reach the level achieved by the IPC-based intra-disk redundancy scheme, which is insensitive to the workload. More specifically, for the case of SATA drives considered, the IPC-based intra-disk redundancy scheme essentially achieves the same reliability as that of a system operating without unrecoverable sector errors, but requires a small increase in capacity, on the order of 6%, for storing the same amount of user data. For heavy workloads, the reliability achieved by scrubbing can be significantly less than that of the intra-disk redundancy scheme. Furthermore, the associated penalty of disk scrubbing on the I/O performance can be significant, whereas that of the IPC-based intra-disk redundancy scheme is minimal.

12. 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, June 2007 (*Proc. ACM SIGMETRICS 2007*, San Diego, CA).
- [2] M. Baker, M. Shah, D. S. H. Rosenthal, M. Roussopoulos, P. Maniatis, T. Giuli, and P. Bungale. A fresh look at the reliability of long-term digital storage. In *Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys 2006)* (Leuven, Belgium), pages 221–234, Apr. 2006.
- [3] M. Blaum, J. Brady, J. Bruck, and J. Mennon. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. *IEEE Trans. Comput.*, 44(2):192–202, Feb. 1995.
- [4] P. M. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson. RAID: High-performance, reliable secondary storage. *ACM Computing Surveys*, 26(2):145–185, June 1994.
- [5] P. Corbett, R. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar. Row-diagonal parity for double disk failure correction. In *Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST)* (San Francisco, CA), pages 1–14, Mar.-Apr. 2004.
- [6] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. 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, June 2006 (*Proc. ACM SIGMETRICS 2006/Performance 2006*, Saint Malo, France).
- [7] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. Rao. A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors. *ACM Trans. Storage*, 4(1), 2008.
- [8] J. G. Elerath and M. Pecht. Enhanced reliability modeling of raid storage systems. In *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (Edinburgh, UK), pages 175–184, June 2007.
- [9] Hitachi Global Storage Technologies, Hitachi Disk Drive Product Datasheets. <http://www.hitachigst.com/>. 2007.
- [10] HP Labs, Private Software. http://tesla.hpl.hp.com/private_software/. 2006.
- [11] LeCroy, Data Storage Solutions, DDNA. <http://www.lecroy.com/tm/solutions/datastorage/DDNA/>. 2007.
- [12] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Chicago, IL), pages 109–116, June 1988.
- [13] 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.
- [14] A. Riska and E. Riedel. Disk drive level workload characterization. In *Proceedings of the USENIX Annual Technical Conference* (Boston, MA), pages 97–102, June 2003.
- [15] C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3):17–28, Mar. 1994.
- [16] D. C. Sawyer. Dependability analysis of parallel systems using a simulation-based approach. *NASA-CR-195762*, Feb. 1994.
- [17] 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.
- [18] 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.

- [19] S. Shah and J. G. Elerath. Reliability analysis of disk drive failure mechanisms. In *Proceedings of the 51th IEEE Annual Reliability and Maintainability Symposium (RAMS)* (Washington, DC), pages 226–231, Jan. 2005.
- [20] The DiskSim Simulation Environment (Version 3.0) <http://www.pdl.cmu.edu/DiskSim/>. 2007.

APPENDIX

A. DETERMINISTIC SCRUBBING MODEL

Proof of Proposition 1:

Let us consider an arbitrary (tagged) sector, denoted by SEC , of a disc. According to the definitions given in Table 1, the probability of an unrecoverable sector error due to a write operation on SEC is equal to P_w . Furthermore, $P_s(t)$ denotes the probability of an unrecoverable error on SEC at time t , and P_s the probability of an unrecoverable error on SEC at an arbitrary point in time. It is now assumed, without loss of generality, that SEC is scrubbed at $t = 0$, such that

$$P_s(0) = 0. \quad (17)$$

We proceed by assuming that the process according to which sector SEC is read/written is Poisson with parameter h given by (7). Consequently, the process according to which sector SEC is read and the process according to which sector SEC is written are Poisson with parameters h_r and h_w , respectively, given by

$$h_r = (1 - r_w)h, \quad \text{and} \quad h_w = r_w h, \quad (18)$$

Thus, the probability q_r that SEC is read within an infinitesimal interval dt is given by

$$q_r \triangleq P(SEC \text{ is read within } dt) = h_r dt. \quad (19)$$

If there is an unrecoverable error on SEC , the read operation will detect it and SEC will be subsequently corrected. Similarly, the probability q_w that SEC is written within an infinitesimal interval dt is given by

$$q_w \triangleq P(SEC \text{ is written within } dt) = h_w dt. \quad (20)$$

According to our assumptions, with probability P_w the write operation will cause an unrecoverable error on SEC . Conditioning on these events, we obtain the probability of an unrecoverable error on SEC at time $t + dt$ as a function of the probability of an unrecoverable error on SEC at time t as follows:

$$\begin{aligned} P_s(t + dt) &= P(SEC \text{ written erroneously} \mid SEC \text{ written within } dt) \\ &\quad P(SEC \text{ written within } dt) \\ &+ P(SEC \text{ erroneous after read} \mid SEC \text{ read within } dt) \\ &\quad P(SEC \text{ read within } dt) \\ &+ P(SEC \text{ erroneous at } t + dt \mid SEC \text{ not r/w within } dt) \\ &\quad P(SEC \text{ not read or written within } dt). \end{aligned} \quad (21)$$

From the above, and using (19) and (20), (21) yields

$$P_s(t + dt) = P_w q_w + 0 q_r + P_s(t) (1 - q_r - q_w), \quad (22)$$

or

$$\frac{P_s(t + dt) - P_s(t)}{dt} = \frac{q_w P_w - (q_r + q_w) P_s(t)}{dt}. \quad (23)$$

By taking the limit $dt \rightarrow 0$, and using (18), (19), and (20), (23) yields

$$P'_s(t) \triangleq \lim_{dt \rightarrow 0} \frac{P_s(t + dt) - P_s(t)}{dt} = h_w P_w - h P_s(t). \quad (24)$$

Solving the differential equation (24) for $P_s(t)$, and using (17) and (18) yields

$$P_s(t) = r_w P_w (1 - e^{-ht}), \quad \text{for } 0 \leq t < T_s. \quad (25)$$

Considering the periodicity T_s of the scrubbing process, and using (10), (25) yields

$$P_s(t) = \left(1 - e^{-h(t \bmod T_s)}\right) P_e, \quad \text{for } t \in \mathbb{R}. \quad (26)$$

Let us now consider an arbitrary point in time, t , and define $\omega \triangleq t \bmod T_s$, which implies that ω is uniformly distributed in the interval $[0, T_s)$. Thus,

$$P_s = \frac{1}{T_s} \int_0^{T_s} P_s(\omega) d\omega. \quad (27)$$

Substituting (26) into (27), after some manipulations, yields (9). \square

B. RANDOM SCRUBBING MODEL

Proof of Proposition 2:

Although the scrubbing model considered here is different from that considered in [18], it turns out that the result sought can be derived based on that obtained in [18], under the assumption that sectors are written at random and according to a Poisson process. More specifically, the parameter τ used in [18, Sec. 1] represents the rate at which a sector gets corrected, which, in that case, is equal to the scrubbing rate. According to the model considered here, however, a sector with an unrecoverable error gets corrected by either a scrubbing operation, a read operation, or a successful write operation. The rate of these events are $1/T_s$, h_r , and $h_w(1 - P_w)$, respectively. Consequently, in our case $\tau = 1/T_s + h_r + h_w(1 - P_w)$. Also, the parameter λ_{bf} used in [18] expresses the rate at which a disk block gets corrupted. Here, considering a sector rather than a block, the rate at which a sector gets corrupted is equal to the product of h_w , the rate at which a given sector is written, and P_w , the probability that the write operation results in an unrecoverable sector error.

For the case of random scrubbing, the probability of failure derived in [18, Sec. 4.1] is given by $1/(1 + \tau/\lambda_{bf})$. By making the following substitutions $\tau = 1/T_s + h_r + h_w(1 - P_w)$ and $\lambda_{bf} = h_w P_w$ into this expression, and using (10) and (18), we obtain (11). \square