Improving Disk Array Reliability Through Expedited Scrubbing

Jehan-François Pâris
Computer Science Dept.
University of Houston
Houston, TX 77204
paris@cs.uh.edu

Thomas Schwarz, S. J.
Depto. de Informática y
Ciencias de la Computación
Universidad Católica del Uruguay
11600 Montevideo, Uruguay
tschwarz@calprov.org

Ahmed Amer
Computer Eng. Dept.
Santa Clara University
Santa Clara, CA 95053
a.amer@acm.org

Darrell D. E. Long[1]
Computer Science Dept.
University of California
Santa Cruz, CA 95064
darrell@cs.ucsc.edu

***Abstract*—Disk scrubbing periodically scans the contents of a disk array to detect the presence of irrecoverable read errors and reconstitute the contents of the lost blocks using the built-in redundancy of the disk array. We address the issue of scheduling scrubbing runs in disk arrays that can tolerate two disk failures without incurring a data loss, and propose to start an urgent scrubbing run of the whole array whenever a disk failure is detected. Used alone or in combination with periodic scrubbing runs, these expedited runs can improve the mean time to data loss of disk arrays over a wide range of disk repair times. As a result, our technique eliminates the need for frequent scrubbing runs and the need to maintain spare disks and personnel on site to replace failed disks within a twenty-four hour interval.**

Keywords-irrecoverable read errors; RAID arrays; disk scrubbing.

I. INTRODUCTION

As today's disks are much larger than they were five to ten years ago, the probability of encountering one or more bad blocks on a given disk is much higher now than it was then. This situation has now reached a point where irrecoverable read errors are a major cause of data losses in disk arrays. For instance, we have ample anecdotal evidence that more failures of RAID level 5 arrays [5, 11, 16, 21] are caused by a combination of a single disk failure and irrecoverable read errors on one or more of the surviving disks than by the simultaneous failure of two disks [12].

Two complementary approaches can be used to protect data against irrecoverable read errors. First, we can mitigate the effects of these errors by using a more robust disk array organization. One of the recommended options consists of switching to disk array organizations, such as RAID level 6 [4, 22] that can tolerate either two simultaneous disk failures or a single disk failure and bad blocks on one or more of the remaining disks. A more recent solution consists of adopting an intradisk parity scheme that introduces an additional level of redundancy inside each disk [6, 23]. Second, we can periodically scan the disk contents to detect irrecoverable read errors and attempt to recover the lost blocks by periodically "scrubbing" the disk contents [3, 26].

Consider for instance the case of a RAID level 6 array with eight to twelve disks. The array will probably use disk scrubbing as an additional precaution against data loss. The critical decision then is the frequency of the disk scrubbing runs. While frequent scrubbing will reduce the number of irrecoverable read errors, they will also accelerate the disk aging process thus causing additional errors.

We argue that we should consider the timing of the scrubbing runs as carefully as their frequency. Observe first that a recently scrubbed disk is error-free and will remain in that state until new bad blocks manifest themselves. This is to say that the most effective scrubbing scans are those that were completed immediately before one of the array disks fails.

We cannot effectively predict disk failures but we can predict when a disk failure is likely to result in a data loss. In the case of a RAID level 6, it would be a second disk failure occurring while uncorrected irrecoverable read errors are present on one or more of the remaining disks. Hence the best way to protect the data against that failure is to run without any delay a scrubbing scan of all the disks in the array. We call this technique *expedited scrubbing* as the scrubbing run has to be performed as quickly as possible to reduce the window of vulnerability of the array.

Our preliminary results indicate that expedited scrubbing can significantly increase the mean time to data loss (MTTDL) of RAID level 6 and alleviate the need for frequent scrubbing runs. In addition, the technique achieves good MTTDLs over a wide range of disk repair times thus alleviating the need for maintaining a local pool of spare disks and having maintenance personnel on call around the clock.

The remainder of this paper is organized as follows. Section II reviews previous work. Section III introduces our technique while Section IV discusses its performance. Section V mentions how our solution could be applied to other fault-tolerant disk array organizations and Section VI has our conclusions.

II. IRRECOVERABLE READ ERRORS

Irrecoverable read errors are also known as bad blocks or latent sector errors because the data loss is not detected until the block is accessed. They are particularly harmful when they occur during the data reconstruction phase of a RAID level 5 array that has one failed disk, as they result in a data loss [12].

[1] Supported in part by the Petascale Data Storage Institute under Department of Energy Award DE-FC02-06ER25768.

Disk drives fail for a number of reasons [1, 8, 17–20]. Some failures result in losing access to a complete disk. These full disk failures occur often without warning, but are sometimes preceded by a burst of recoverable or irrecoverable sector errors.

Other disk failures, known as latent failures, only affect a single sector or a relatively small number of sectors. Bairavasundaram and his colleagues from the University of Wisconsin-Madison found that 3.45% of 1.53 million disks studied developed latent sector errors over a period of 32 months, though the number of affected sectors tended to be small [1]. There are numerous mechanisms that can cause such errors [10]: Many latent sector errors are caused during writing such as "high-fly writes", writing on damaged parts of the media caused by loose hard particles, pits and voids, or off-track writes. These errors can be discovered by validating writes shortly after reads. Even after a successful write, thermal asperities, surface scratches by hard particles, smears by "soft" particles (such as stainless steel), or corrosion can occasion data losses. Remarkably, "bit-rot", that is, the degradation of the magnetization pattern in the media, is currently not a significant cause. These errors are latent, they are only recognized when trying to read the sector.

The redundant storage organizations used by most multi-disk storage systems—such as RAID levels 5 and 6—allow recovery from these latent errors in the absence of any other disk failures. Frequently, the affected sectors are marked by the internal disk controller as unusable and their logical block address is mapped to spare sectors. However, a system can lose data if it recovers from a complete disk failure relying on a disk with latent sector errors. Two possibilities arise to forestall this danger. First, we can use additional redundancy, either by using a two-or-more failure resilient redundancy scheme such as RAID Level 6 or by using additional redundancy in the disk itself (Intra-disk Redundancy [6, 26]). Second, latent disk errors can be detected and dealt with before they make recovery from a complete hard drive failure impossible. This latter approach is known as disk scrubbing [3, 12, 22, 25]. Media scrubs can use the SCSI VERIFY command to validate a sector's integrity using the error control coding embedded in the storage of sectors on magnetic media. It is possible, though usually not necessary, to further verify the contents by maintaining and recalculating a checksum over a number of disk sectors [27], since the number of instances where a sector failure changes contents erroneously are extremely rare.

Scrubbing identifies latent sector errors by systematically checking all sectors for readability. Frequent scrubbing greatly increases the probability that a disk does not contain latent sectors, but has a negative impact on performance (since seeks cannot be preempted). Additionally, by increasing the disk load, it can increase latent sector error and hard drive failure rates. To further improve its effectiveness, scrubbing can be supplemented with "read after write" to eliminate a category of causes for latent errors. Scrubbing can be deterministic or opportunistic. Deterministic scrubbing schedules scrubbing operations at fixed intervals and guarantees a full sweep of the disk within a set period. In contrast, opportunistic scrubbing uses periods of zero user utilization to perform its task.

Baker *et al.* [2] enumerated the multiple threats to data survivability in disk arrays and presented a window of vulnerability model (WOV) that takes into account temporality of faults. They used that model to evaluate the impact of latent block errors on the MTTDL of mirrored data and based their estimate of the frequency of latent block errors on manufacturers' specification of a 2^{-14} worst-case irrecoverable bit rate.

Elerath identified major categories of disk failure modes and discussed the impact of latent defects [7]. He observed that read error rates had actually decreased between 2004 and 2007 and reported read-error rates varying between 3.2×10^{-13} and 8×10^{-15} errors/byte. In a more recent study [6], he introduces a formula that provides a good approximation of the expected number of data losses caused by double disk failures for an $N + 1$ RAID array. His model takes into account latent disk errors that reveal themselves during the array rebuilding process and assumes that disk failures obey a Weibull distribution.

III. Expedited Scrubbing

As we mentioned earlier, recently scrubbed disks are error-free and remain in that state until new bad blocks manifest themselves. Thus the most efficient time to schedule a scrubbing run is just before the disk contents are needed.

Many recent disk organizations tolerate two or more disk failures without data loss in the absence of irrecoverable read errors. Returning to the case of a RAID level 6 array, we can see that this organization can tolerate:

a) Two disk failures in the absence of irrecoverable read errors.
b) One disk failure and an arbitrary number of irrecoverable read errors, as long as each parity stripe contains at most *one* bad block.
c) An arbitrary number of irrecoverable read errors as long as each parity stripe contains at most *two* bad blocks.

Given the fairly low percentage of bad blocks in infected disk drives, we can safely assume that a RAID level 6 array will nearly always remain operational after having lost a disk but will lose data if a second disk fails unless all irrecoverable read errors have been corrected beforehand. To address that issue, we propose to start an urgent scrubbing scan whenever we detect a failure of one of the array disks. We call this technique *expedited scrubbing* because this scrubbing must be performed as expeditiously as possible in order to bring the array into a safer state as quickly as possible. As we can see, our technique can be used either alone or in combination with periodic scrubbing.

Expedited scrubbing belongs to the realm of *reactive approaches* to fault-tolerance because it takes no action in the absence of any disk failures. As a result, it consumes no resources when the array operates correctly and is only activated when it is most needed.

As most techniques, expedited scrubbing has its limitations. First, it does not apply to RAID level 5 arrays and other array organizations that can only tolerate a single disk failure. Second, it assumes that the scrubbing process can be performed relatively quickly, say, in 24 hours or less, without disturbing the operation of the array. Finally, it does not perform well in the presence of correlated disk failures. Consider for instance the case of an overheated disk array that has just lost a disk. Initiating a scrubbing run at that moment will result in the production of additional heat and may precipitate the failure of other disks.

IV. Performance Analysis

We will base our study on the observations of Bairavasundaram *et al.* because they were collected from a very large disk population and offer us a better insight into the effect of bad blocks on disk array reliability.

Consider for instance the case of a RAID level 5 array with $N + 1$ disks. We know that the array can reconstitute missing data as long as each block stripe participating in the reconstruction process contains only one block that cannot be read. This is why such arrays can tolerate a single disk failure but not a double disk failure nor the combination of a single disk failure with one or more bad blocks on one of the N remaining disks. Let us now consider what happens when all the array disks are operational but two of them contain bad blocks. Thanks to the built-in redundancy of the array, we will be able to reconstitute all the lost data as long as no block stripe contains more than one bad block.

Assuming that a fraction f of the blocks of a disk are bad, the probability of observing more than one bad block in a single stripe will be

$$n_b(1-(1-f)^{N+1}-(N+1)f(1-f)^N),$$

where n_b is the number of stripes in the array (and thus the number of blocks on each disk) and $N + 1$ its number of disks.

Consider now the case of a RAID array consisting of five one-terabyte disks with a block size of four kilobytes. The formula above shows that we would need to observe at least 495 bad blocks on each disk to have a one percent probability of having two bad blocks in the same stripe. Given that Bairavasundaram *et al.* observed an infection rate of 3.54 percent over 32 months and reported that 80 percent of the affected disks had less than 50 bad blocks, we can safely infer that the occurrence of two bad blocks in the same stripe will be an extremely unlikely event whose probability can be safely neglected.

A. Fundamental assumptions

Estimating the reliability of a storage system means estimating the probability $R(t)$ that the system will operate correctly over the time interval $[0, t]$ given that it operated correctly at time $t = 0$. Computing that function requires solving a system of linear differential equations, a task that becomes quickly unmanageable as the complexity of the system grows. A simpler option is to focus on the mean time to data loss (MTTDL) of the storage system, which is the approach we will take here.

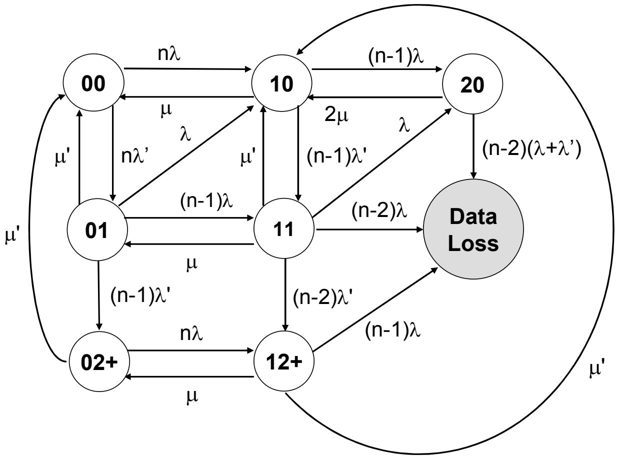

Fig. 1. State-probability transition diagram for a RAID level 6 array consisting of *n* disks, all subject to disk failures and irrecoverable read errors.

Our system model consists of an array of disks with independent failure modes. When a disk fails, a repair process is immediately initiated for that disk. Should several disks fail, the repair process will be performed in parallel on those disks. We assume that disk failures are independent events that are exponentially distributed with mean λ. In addition, we require repairs to be exponentially distributed with mean μ. Both hypotheses are necessary to represent each system by a Markov process with a finite number of states.

We assume that each disk is initially in a state where all its defective blocks have been "mapped out" in a way that prevents users from accessing them. We further assume that the appearances of bad blocks on disks are independent events and are exponentially distributed with mean λ'. This is to say that we are not modeling the formation of individual bad blocks but rather the transition from a state where all defective blocks have been mapped out to a state where the disk has one or more bad blocks holding data. Once a disk has one or more bad blocks, it remains in that state until the missing data are reconstructed and written in one of the spare sectors of the disk. This recovery could be the result of a failed read access or a periodic scrubbing of the whole disk. Combining the effect of these two processes we assume that disks with bad blocks will return to their original state at a rate μ' and that these transitions follow an exponential law.

B. A RAID level 6 *array using periodic scrubbing*

Let us consider the case of a RAID level 6 with *n* disks [4, 22] whose contents are periodically scrubbed to detect and repair irrecoverable read errors. As Fig. 1 shows, the array can be at any time in one out of eight possible states.

State <00> represents the initial state of the system when the *n* disks are operational and all their bad blocks are mapped out. The first disk failure will bring the array to state <10>. A second disk failure will bring it to state <20> and a third disk failure will result in a data loss.

The presence of bad blocks on a single disk will respectively bring the array from state <00> to state <01>, from state <10> to state <11> and from state <20> to the failed state. Similarly, the presence of bad blocks on a second disk would respectively move the array from state <01> to state <02+> and from state <11> to state <12+>.

The appearance of bad blocks on a third disk will leave the system in either state <02+> or <12+> as we do not distinguish between array configurations having bad blocks on two disks and those having bad blocks on three or more disks.

From state <01>, the array can:

a) Go to state <10> if the disk that has bad blocks fail;

b) Go to state <11> if any of the $n - 1$ other disks fail;

c) Go to state <02+> if more disks develop bad blocks;

d) Return to its original state once all bad blocks have been mapped out.

From state <11>, the array can:

a) Go to state <20> if the disk that has bad blocks fails;

b) Incur a data loss if any of the $n - 2$ other disks fails;

c) Go to state <12+> if one or more disks develop bad blocks;

d) Return to its original state once all bad blocks have been mapped out.

Observe that our model assumes that a disk failure occurring when the array is in state <02+> will always bring that array into state <12+>. As a result, we neglect the less frequent case where the array has bad blocks on exactly two disks and one of them fails, thus bringing the array into state <11>. This observation does not apply to state <12+> since the loss of any of its $n - 1$ operational disks will result in a data loss.

Replacing the failed disks will bring the array from state <20> to state <10> and from state <10> to its original state. In the same way, scrubbing scans will bring the array from state <02+> to state <00> and from state <12+> to state <10>.

The Kolmogorov system of differential equations describing the behavior of the disk array is

$$\frac{dp_{00}(t)}{dt} = -n(\lambda+\lambda')p_{00}(t) + \mu p_{10}(t) + \mu'(p_{01}(t) + p_{02+}(t)),$$

$$\frac{dp_{01}(t)}{dt} = -(n\lambda + (n-1)\lambda' + \mu')p_{01}(t) + n\lambda' p_{00}(t) + \mu p_{11},$$

$$\frac{dp_{02+}(t)}{dt} = -(n\lambda + \mu')p_{02+}(t) + (n-1)\lambda' p_{01}(t) + \mu p_{12+}(t),$$

$$\frac{dp_{10}(t)}{dt} = -((n-1)(\lambda+\lambda') + \mu)p_{10}(t) + n\lambda p_{00}(t) + \lambda p_{01}(t) + 2\mu p_{20} + \mu'(p_{11}(t) + p_{12+}(t)),$$

$$\frac{dp_{11}(t)}{dt} = -((n-1)\lambda + (n-2)\lambda' + \mu + \mu')p_{11}(t) + (n-1)\lambda p_{01}(t) + (n-1)\lambda' p_{10}(t),$$

$$\frac{dp_{12+}(t)}{dt} = -((n-1)\lambda + \mu + \mu')p_{12+}(t) + n\lambda p_{02+}(t) + (n-2)\lambda' p_{11}(t),,$$

$$\frac{dp_{20}(t)}{dt} = -((n-2)(\lambda+\lambda') + 2\mu)p_{20}(t) + (n-1)\lambda p_{10}(t) + \lambda p_{11}(t).$$

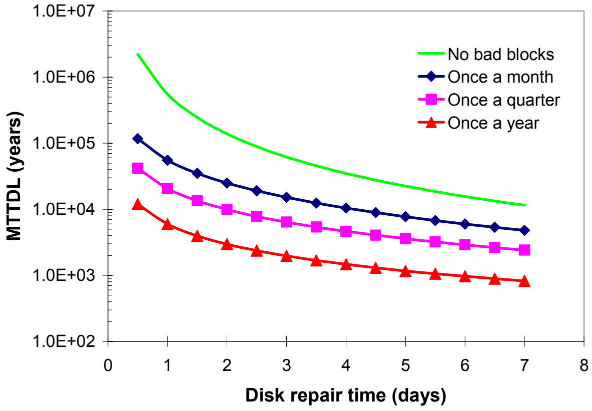

Fig 2. Expected MTTDL of a RAID level 6 array consisting of ten disks, all subject to disk failures and irrecoverable read errors. Each but the top-most curve corresponds to a specific scrubbing interval.

where $p_{ij}(t)$ is the probability that the system is in state <*ij*> with the initial conditions $p_{00}(0) = 1$ and $p_{ij}(0) = 0$ otherwise. The Laplace transforms of these equations are

$$sp^*_{00}(s) - 1 = -n(\lambda+\lambda')p^*_{00}(s) + \mu p^*_{10}(s) + \mu'(p^*_{01}(s) + p^*_{02+}(s)),$$

$$sp^*_{01}(s) = -(n\lambda + (n-1)\lambda' + \mu')p^*_{01}(s) + n\lambda' p^*_{00}(s) + \mu p^*_{11}(s),$$

$$sp^*_{02+}(s) = -(n\lambda + \mu')p^*_{02+}(s) + (n-1)\lambda' p^*_{01}(s) + \mu p^*_{12+}(s),$$

$$sp^*_{10}(s) = -((n-1)(\lambda+\lambda') + \mu)p^*_{10}(s) + n\lambda p^*_{00}(s) + \lambda p^*_{01}(s) + 2\mu p^*_{20}(s) + \mu'(p^*_{11}(s) + p^*_{12+}(s)),$$

$$sp^*_{11}(s) = -((n-1)\lambda + (n-2)\lambda' + \mu + \mu')p^*_{11}(s) + (n-1)\lambda p^*_{01}(s) + (n-1)\lambda' p^*_{10}(s),$$

$$sp^*_{12+}(s) = -((n-1)\lambda + \mu + \mu')p^*_{12+}(s) + n\lambda p^*_{02+}(s) + (n-2)\lambda' p^*_{11}(s),$$

$$sp^*_{20}(s) = -((n-2)(\lambda+\lambda') + 2\mu)p^*_{20}(s) + (n-1)\lambda p^*_{10}(s)+, \lambda p^*_{11}(s).$$

Observing that the mean time to data loss (MTTDL) of the disk array is

$$MTTDL = \sum_{i,j} p^*_{ij}(0),$$

we solve the system of Laplace transforms for $s = 0$ and compute the MTTDL of the array. The result is a quotient of two very large polynomials of degree 6 for the numerator and of degree 7 for the denominator.

When $\mu' \to \infty$, that expression simplifies into the traditional formula for RAID level 6 arrays

$$MTTDL = \frac{(3n^2 - 6n + 2)\lambda^2 + 3(n-2)\lambda\mu + 2\mu^2}{n(n-1)(n-2)\lambda^3}.$$

Fig. 2 displays the MTTDLs achieved by a RAID level 6 array with ten disks for selected values of the scrubbing interval and repair times that vary between half a day and seven days. It confirms the results of a previous study that

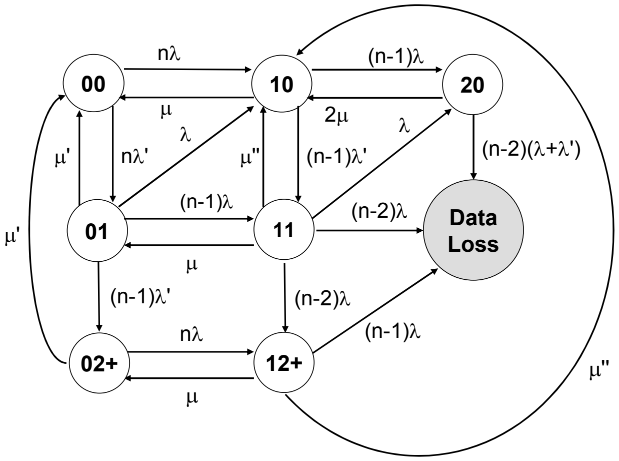

Fig. 3. State-probability transition diagram for a RAID level 6 array consisting of *n* disks, all subject to disk failures and irrecoverable read errors, when we allow expedited scrubbing.

investigated a RAID level 6 array with six disks and concluded that scheduling scrubbing scans once a month instead of once a year can increase the MTTDL of the array from 600 to almost 1,000 percent depending on the disk repair time [15].

C. *Enabling expedited scrubbing*

Let us now investigate what impact expedited scrubbing would have on the MTTDL of the array. We assume now that we initiate an accelerated scrubbing procedure at a rate μ" much higher than the regular scrubbing rate whenever we detect the loss of a disk. As Fig. 3 shows, the state transition probability diagram of our array is barely distinguishable from that of an array undergoing periodic scrubbing. The sole difference is that the state transitions from states <11> and <12+> now occur at a rate μ" that is measured in hours instead of months.

Using the same techniques as in the previous case, we compute the MTTDL of a RAID level 6 array with ten disks that undergo expedited scrubbing each time it detects the loss of a disk. The result is an even larger quotient of two very large polynomials of degree 6 for the numerator and of degree 7 for the denominator.

Fig. 4 displays the MTTDLs achieved by a RAID level 6 array with ten disks for selected values of the scrubbing interval and repair times assuming that expedited scrubbing can be done in 24 hours. As we can see, the MTTDL improvements resulting from expedited scrubbing greatly vary with the scrubbing interval and the repair times. Emergency disk scrubbing performs best at the longest scrubbing interval and the longest repair times. Conversely, emergency disk scrubbing offers much less benefits when the disks are frequently scrubbed and quickly repaired. This should not surprise us. First, frequently scrubbed disks are less likely to contain bad blocks. Second, fast repair times leave less time for the beneficial effects of expedited scrubbing to manifest themselves. Overall, one of the major advantages of expedited scrubbing is that it performs well over a wide range of

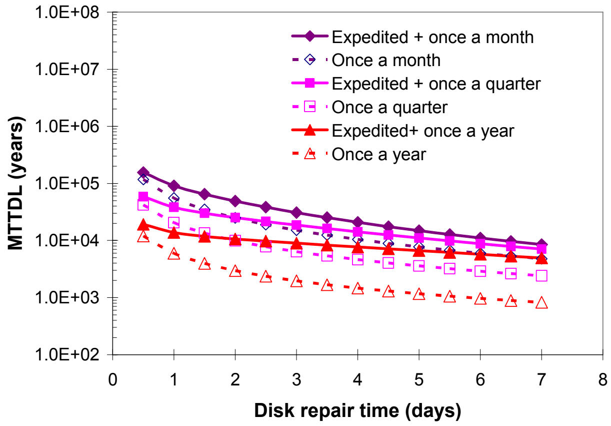

Fig 4. Expected MTTDL of a RAID level 6 array consisting of ten disks, assuming that its disks are scrubbed at regular time intervals and scrubbed within a twenty-four hour interval after a disk failure.

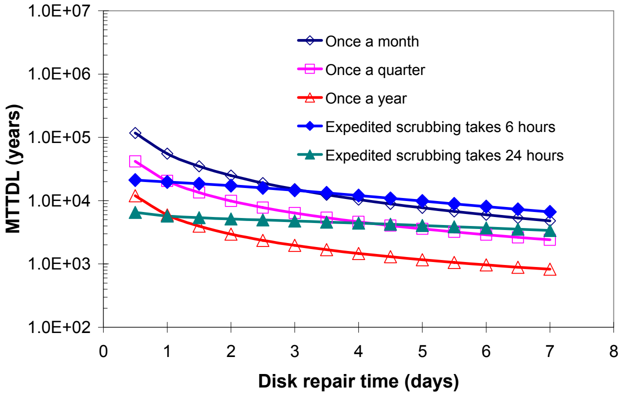

Fig 5. Expected MTTDL of a RAID level 6 array consisting of ten disks, using either periodic scrubbing or pure expedited scrubbing.

disk repair times. This is a significant advantage because fast repair times require maintaining a local pool of spare disks and having staff on call round the clock. Since RAID level 6 arrays using expedited scrubbing tolerate repair times of up to one week, if not more, they will be cheaper and easier to maintain than arrays relying on periodic scrubbing alone.

D. *Relying exclusively on expedited scrubbing*

To test the limits of our approach, we considered the option of relying exclusively on expedited scrubbing, thus completely eliminating periodic disk scrubbings. This approach is particularly suited to large archival systems that experience long periods of idleness during which their disks remain powered down and need to be powered up to be scrubbed

Fig. 5 displays the MTTDLs achieved by a RAID level 6 array with ten disks for selected values of the expedited scrubbing time and the disk repair time. These values are fairly pessimistic because they were obtained by positing μ' = 0 in our previous model. In reality, disks have a finite lifetime and irrecoverable read errors will get automatically repaired every time we transfer its data to a new disk array.

As we can see, exclusively relying on expedited scrubbings results in fairly good MTTDLs as long as these scrubbings can be performed fairly quickly, say, in less than 24 hours after a disk failure. We also notice that pure expedited scrubbing performs quite well over a wide range of disk repair times.

E. Discussion

In order to be able to use a stochastic model with a finite number of states, we had to introduce several assumptions that are not valid for real systems. First, we assumed that failure occurrences and repair times were exponentially distributed. This is not true for real disk populations as failures tend to be distributed according to a Weibull distribution and repair time distributions have much smaller coefficients of variation than the exponential distribution. Second, we assumed constant failure rates λ and λ' over the lifetime of the array, while actual failure rates tend to decrease over the first few months of the disk lifetime and increase again after a few years.

Other simplifying assumptions were the result of a lack of data. First, the failure rates λ and λ' that we selected were average rates estimated over very large heterogeneous populations of disks comprising both enterprise class and nearline class disks. Second, we did not take into account the impact of correlated failures on the MTTDL of the array. This could have been handled by making the two failure rates λ and λ' functions $\lambda(m)$ and $\lambda'(m)$ of the number m of previous failures. We could not do this due to insufficient data.

A last issue concerns our choice of MTTDL to represent the reliability of disk arrays. MTTDLs characterize fairly well the behavior of disk arrays that would remain in service until they fail without ever being replaced for any reason other than a device failure. This is rarely the case as disk arrays are typically replaced after five to seven years, that is, well before they experience any failure. MTTDLs do not take into account this relatively short lifetime, and tend to overestimate the probability of a data loss over their lifetime. This effect remains negligible as long as the time to repair an individual disk is at least one thousand times shorter than its MTTF [14].

V. Application to other fault-tolerant disk organizations

While we have only considered so far the application of our technique to RAID level 6 arrays, it applies to all fault-tolerant disk organizations tolerating two or more disk failures without data loss. In the case of organizations tolerating more than two data losses, such as HDD-1 and HDD-2 schemes [24], we may want to adjust the priority level of the scrubbing tasks to their urgency. As an example, the failure of a single disk might trigger a low-to-medium priority scrubbing process that would not affect other disk requests while a second disk failure would increase the priority of the scrubbing process as data preservation becomes our paramount concern.

VI. Conclusion

We have presented a new scrubbing technique for disk arrays tolerating two or more disk failures. Unlike periodic scrubbing or opportunistic scrubbing, expedited scrubbing starts an urgent scrubbing of the whole array whenever a disk failure is detected. Used alone or in combination with periodic scrubbing, our technique can improve the mean time to data loss of disk arrays over a wide range of disk repair times. As a result, it allows less frequent scrubbing runs and reduces the need to maintain spare disks and personnel on site to replace failed disk drives within 24 hours.

VII. References

[1] L. Bairavasundaram, G. Goodson, S. Pasupathy, J. Schindler, "An analysis of latent sector errors in disk drives," *Proc. ACM SIGMETRICS 2007 International Conference on Measurement and Modeling of Computer Systems*, San Diego, CA, 2007.

[2] M. Baker, M. Shah, D. S. H. Rosenthal, M. Roussopoulos, P. Maniatis, T. J. Giuli, and P. Bungale, "A fresh look at the reliability of long-term storage," *Proc. First EuroSys Conference* (EuroSys 2006), Leuven, Belgium, Apr. 2006.

[3] J. Belsau, A. Permut, G. Rudeseal: "Disk Scrubbing System," US Patent 5,632,012, May 20, 1997.

[4] W. A. Burkhard and J. Menon, "Disk array storage system reliability," *Proc.* 23rd *International Symposium on Fault-Tolerant Computing* (FTCS '93), Toulouse, France, pp. 432–441, June 1993.

[5] P. M. Chen, E. K. Lee, G. A. Gibson, R. Katz, and D. Patterson, "RAID: High-performance, reliable secondary storage," *ACM Computing Surveys*, 26(2):145–185, June 1994.

[6] A. Dholakia , E. Eleftheriou , X.-Y. Hu , Il. Iliadis , J. Menon, K.K. Rao, "A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors," *ACM Transactions on Storage*, 4(1):1-42, May 2008.

[7] J. G. Elerath, "Hard disk drives: The good, the bad and the ugly!" *ACM Queue* 5(6):28–37, Sept./Oct., 2007.

[8] J. G. Elerath, "A simple equation for estimating the reliability of an N+1 redundant array of independent disks (RAID)," *Proc.* 39th *nternational Conference on Dependable Systems and Networks* (DSN 2009), Estoril, Portugal, June 2009.

[9] J. Elerath and S. Shah: "Server class disk drives: How reliable are they?" *Proc. IEEE Annual Reliability and Maintainability Symposium* (RAMS '04), 2004.

[10] J. Elerath and M. Pecht, "Enhanced reliability modeling of RAID storage systems." *Proc.* 37th *IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN '07), Edinburgh, GB, pages 175–184, June 2007.

[11] G. A. Gibson, "Redundant disk arrays: Reliable, parallel secondary storage," *Ph.D. Thesis*, University of California, Berkeley, 1990.

[12] R. Harris, "Why RAID 5 stops working in 2009," ZDnet, http://blogs.zdnet.com/storage/?p=162, July 18, 2007 (Retrieved on March 12, 2010)

[13] H. Kari: "Latent sector faults and reliability of disk arrays", Ph.D. thesis, Helsinki University of Technology, Sep. 1997.

[14] J.-F. Pâris, T. J. E. Schwarz, D. D. E. Long and A. Amer, "When MTTDLs are not good enough: Providing better estimates of disk array reliability," *Proc.* 7th *International Information and Telecommunication Technologies*

Symposium (I2TS '08), Foz do Iguaçu, PR, Brazil, pp. 140–145, Dec. 2008.

[15] J.-F. Pâris, A. Amer, D. D. E. Long and T. J. E. Schwarz, "Evaluating the impact of irrecoverable read errors on disk array reliability," *Proc. IEEE 15th Pacific Rim International Symposium on Dependable Computing* (PRDC '09), Shanghai, China, pp. 379–384, Nov. 2009.

[16] D. A. Patterson, G. A. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," *Proc. SIGMOD 1988 International Conference on Data Management*, Chicago, IL, pp. 109–116, June 1988.

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

[18] S. Shah and J. Elerath, "Reliability analysis of disk drive failure mechanisms", *Proc. 2005 IEEE Annual Reliability and Maintainability Symposium* (RAMS '05), Washington, DC, Jan. 2005.

[19] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *Proc. 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN 2006), Philadelphia, PA, pp. 249–258 , June 2006.

[20] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" *Proc. 5th USENIX Conference on File and Storage Technologies* (FAST '07), San Jose, CA, pp. 1–16, Feb. 2007.

[21] M. Schulze, G. Gibson, R. Katz and D. Patterson, "How reliable is a RAID?" *Proc. Spring COMPCON '89 Conference*, San Francisco, CA, pp. 118–123, March 1989.

[22] T. J. E. Schwarz and W. A. Burkhard. "RAID organization and performance," *Proc. 12th International Conference on Distributed Computing Systems* (ICDCS '92), Yokohama, Japan, pp. 318–325 June 1992.

[23] 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," *Proc. 12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (MASCOTS '04), pp. 409–418, Oct. 2004.

[24] C.-S. Tau and T.-I Wang, "Efficient parity placement schemes for tolerating triple disk failures in RAID architectures," *Proc. 17th International Conference on Advanced Information Networking and Applications* (AINA'03), Xi'an, China, March 2003.

[25] E. Thereska, J. Schindler, J. Bucy, B. Salmon, C. Lumb, and G. Ganger: "A framework for building unobtrusive disk maintenance applications, *Proc. 3rd USENIX Conference on File and Storage Technologies* (FAST '04), San Francisco, CA, pp. 213–226, Apr. 2004

[26] A. Wildani, T. J. E. Schwarz, E. L. Miller and D. D. E. Long, "Protecting against rare event failures in archival systems," *Proc. 17th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (MASCOTS '09), London, GB, Sep. 2009.

[27] Q. Xin, E. Miller, D. Long, S. Brandt, W. Litwin, and T. J. E. Schwarz, "Selecting reliability mechanism for a large object-based storage system," *Proc. 20th Symposium on Mass Storage Systems and Technologies* (MSST '03), San Diego, CA, pp. 146–156, Apr. 2003.