UC Santa Cruz

Title
Protecting Data Against Early Disk Failures

Permalink
https://escholarship.org/uc/item/357320k4

Authors
Paris, Jehan-Francois
Schwarz, Thomas JE
Long, Darrell

Publication Date
2006-12-06

Copyright Information

Peer reviewed

Protecting Data Against Early Disk Failures

Jehan-François Pâris
Dept. of Computer Science
University of Houston
Houston, TX 77204-3010

Thomas J. E. Schwarz
Dept. of Computer Engineering
Santa Clara University
Santa Clara, CA 95053

Darrell D. E. Long[1]
Dept. of Computer Science
University of California
Santa Cruz, CA 95064

Abstract—**Disk drives are known to fail at a higher rate during their first year of operation than during the remaining years of their useful lifetime. We propose to use the free space that normally exists on new disks to minimize the risk of data loss during that first year. Our technique applies to disk arrays that mirror their data on two disks. Whenever a disk fails, the array will reorganize itself by storing a new copy of the disk that failed on one or more disks that have free space. This will protect the data against any single disk failure until the failed disk gets replaced and the system reverts to its original state. A Markov analysis of the behavior of a small system consisting of two pairs of mirrored disks indicates that our technique can reduce the probability of a data loss during the first year of operation of the system by at least 75 percent provided the disks have 34 percent of spare space.**

I. INTRODUCTION

An ever increasing number of organizations now rely on disk arrays for the long-term storage of their data. This trend results from the convergence of several factors. First, advances in magnetic storage technology have considerably reduced the cost of storing data online. Second, regulatory requirements now obligate public corporations to retain their audit data over longer periods of time than in the past and to keep them immediately accessible. Finally, the rate at which digital data are produced keeps increasing in nearly all organizations [5].

A main challenge facing the designer of a storage system is how to ensure the survival of its data over periods that can span decades. Mirroring and erasure codes are the two preferred techniques to achieve this goal. Mirroring maintains multiple redundant copies of the stored data while *m*-out-of-*n* codes store data on *n* distinct disks along with enough redundant information to allow access to the data in the event $n - m$ of these disks fail. The best-known organizations using these codes are RAID level 5, which uses an $(n - 1)$-out-of-*n* code, and RAID level 6, which uses an $(n - 2)$-out-of-*n* code.

Two issues that greatly complicate the selection of the best storage organization for a given application are disk infant mortality and the bad batch problem. Disks have much higher failure rates—between two and three times higher than those

indicated by their mean time to failure—during their first year of operation. In addition, most failures resulting from a bad batch of disks also show up sometimes during that year. The traditional solution of burning in devices before actually using them would not help much since a prudent burn-in period would take one year and use up one fifth to one sixth of the disk lifespan [3, 15]. We are thus forced to use disks drives while they are still in their period of high infant mortality, and are still subject to bad batch failures.

The two default options are either ignoring the issue, thus increasing the risk of a data loss during the first year of operation, or taking into account these higher initial failure rates when selecting a specific storage organization.

A better solution exists. It consists of increasing the resiliency of disk arrays during its first year of operation by taking advantage of the spare space they are likely to have. The simplest way to achieve this goal would be to increase the replication level of the stored data, but it would increase the cost of writing new data or updating existing ones. We propose instead a self-adaptive mirrored organization. When all disks are operational, all data are mirrored on two disks. Whenever a disk fails, the system reorganizes itself, by selecting first one or two disks that have enough spare space and storing on them a new copy of the data of the disk that failed. This third copy will remain in place until the failed disk gets replaced and the system reverts to its original condition.

To evaluate the benefits of this new disk organization, we have analyzed the behavior of a small system consisting of two pairs of mirrored disks using standard Markovian assumptions. Our results indicate that our technique can reduce the probability of a data loss during the first year of operation of the system by at least 75 percent. We also found out that even better results could be achieved by taking advantage of the failure prediction capabilities of the new S.M.A.R.T. disks [4, 9, 13].

The rest of the paper is organized as follows: Section 2 surveys previous relevant work. Section 3 introduces our technique and Section 4 evaluates its performance. Finally, Section 5 has our conclusions.
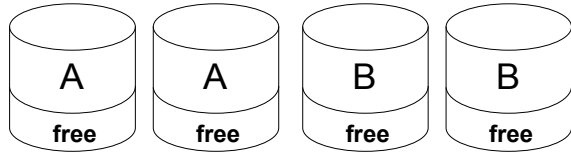

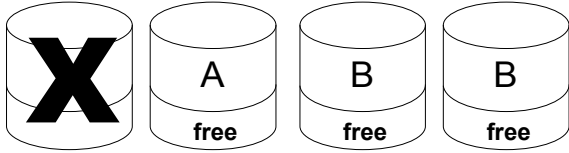Fig. 1. A small disk array consisting of two pairs of mirrored drives.

Fig. 2. The same array .after the failure of one of its four disks.
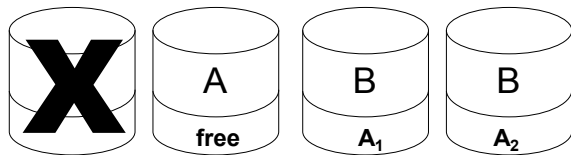

Fig. 3. The same array after a redundant copy of A has been created on two of the remaining drives.

II. PREVIOUS WORK

The idea of creating additional copies of critical data in order to increase their chances of survival is probably as old as the use of symbolic data representations by mankind. Erasure coding appeared first in RAID organizations as $(n-1)$-out-of-n codes [2, 6, 10, 11]. RAID level 6 organizations use $(n-2)$-out-of-n codes to protect data against double disk failures [1].

Much less work has been dedicated to self-organizing fault-tolerant disk arrays. The HP AutoRAID [14] automatically and transparently manages migration of data blocks between a replicated storage class and a RAID level 5 storage class as access patterns change. Its main objective is to save disk space without compromising system performance by storing data that are frequently accessed in a replicated organization while relegating inactive data to a RAID level 5 organization. As a result, it reacts to changes in data access patterns rather than to disk failures.

Sparing is more relevant to our proposal as it provides a form of adaptation to disk failures. Adding a spare disk to a disk array provides the replacement disk for the first failure. Distributed sparing [12] gains performance benefits in the initial state and degrades to normal performance after the first disk failure.

Pâris *et al.* [8] have recently presented a disk array organization that adapts itself to successive disk failures. When all disks are operational, all data are mirrored on two disks. Whenever a disk fails, the array reorganizes itself, by selecting a disk containing redundant data and replacing these
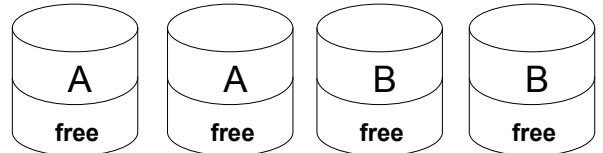

Fig. 4. The same disk array being only half full.

data by their exclusive or (XOR) with the other copy of the data contained on the disk that failed. Once the failed disk is replaced, the array returns to its original configuration. Since this scheme operates by replacing existing data by their XOR, with other data, it does not require any spare space. Its main drawback is a more complex recovery as the data that were overwritten need then to be restored.

The notion of creating new copies of data to replace those lost owing to site failures was first introduced by Pu et al. in the *regeneration algorithm* [7].

III. OUR TECHNIQUE

Our goal is to increase the reliability of storage systems during their first year of operation, period during which they experience higher disk failure rates than during the remainder of their useful lifetime. In addition, we wanted a solution that would not require any additional hardware and would not perturb the normal operation of the storage system. The solution we propose satisfies these two requirements since:

1. It uses the free space that normally exists on recently deployed drives to increase the redundancy of the stored data.
2. It brings no changes to the storage system as long as all disks are operational: new copies of the stored data are only created in response to a disk failure and are deleted as soon as the failed drive has been replaced.

Consider the small disk array displayed on Fig. 1. It consists of two pairs of mirrored disks with data replicated on each pair: data set *A* is replicated on the first pair of disks and data set *B* on the second pair. We will assume that none of the four disks is more than two-thirds full, a reasonable assumption for a disk array that has been recently deployed.

Assume now that one of the disks holding a copy of data set *A*. As shown on Fig. 2, only one remaining copy of that data set remains and the array will become vulnerable to a failure of the disk. Waiting for the replacement of disk A_1 is not an attractive option as the process make take several days. To adapt itself to the failure, the system will immediately locate a pair of disks that do not already contain the data set *A* and store on each of the two disks one half of the data set *A,* thus making the array immune to a single disk failure. Fig. 3 displays the outcome of that reconfiguration. The system will remain in that state until the failed disk gets replaced and the two half copies of data set *A* can be safely discarded.

Consider now the case when none of the four disks is more than half full. As shown on Fig. 4, each disk has enough
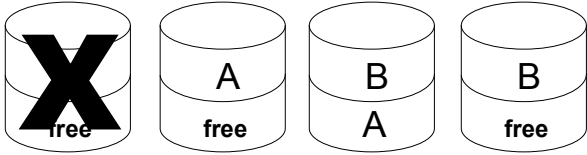
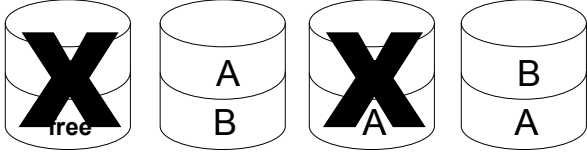Fig. 5. How the array will adapt itself to the loss of a disk.

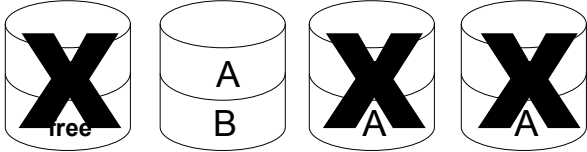Fig. 6. How the array will adapt itself to the loss of a second disk.

Fig. 7. How the array will adapt itself to the loss of a third disk.

Fig. 8. State transition diagram for a self-adaptive array of four drives when none of its drives is more than two-third full.

space to store all the data that are stored on any of the three other disks.

Fig. 5 to 7 represent how the array would reorganize itself after the successive failures of a first, a second and a third drive. As we can see, the array can now tolerate the failure of up to three of the four drives provided that they do not happen in too close succession.

In all cases, the reconfiguration process is totally transparent to the user, who will only observe a reduction of free space after each reconfiguration.

There is one more way to improve the likelihood the data will survive several crashes. Most major drive manufacturers now support to some extent the *Self-Monitoring, Analysis and Reporting Technology* (S.M.A.R.T.), whose purpose is to warn users of impending disk failures [4, 9, 13]. The technique is not perfect: it can only predict approximately 30 percent of hard drive failures since many failures are sudden and unpredictable [9]. Using these warning would allow us to save elsewhere the data that are stored on the disk whose failure was predicted.   Since our technique never overwrites any of the original mirrored copies of the data, such precautionary actions would cause no harm, apart from the additional data traffic they would occasion. This was not the case for a previous proposal for self-adaptive arrays that replaced the contents of one disk by their XOR with the contents of another disk [8].
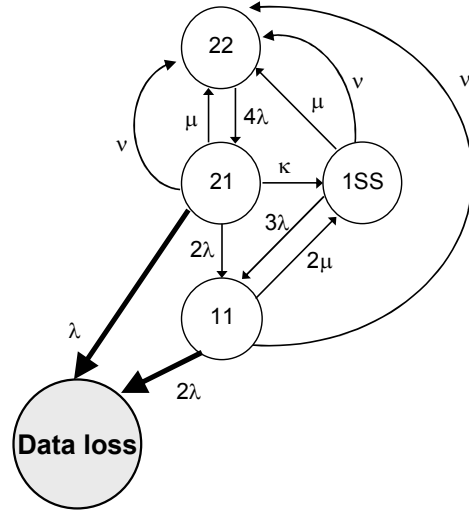
IV. RELIABILITY ANALYSIS

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) and the average failure rate (1/MTTDL) of the system. This is the approach we will take here.

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 drive. Should several disk fail, the repair process will be performed in parallel on those drives.

We assume that disk failures are independent events exponentially distributed with rate λ, and that repairs are exponentially distributed with rate μ. In most cases, most of the repair time will be taken by ordering and scheduling delays while the actual replacement of the failed disk will rarely take more than a few hours. Reorganization transitions corresponding to the creation of additional copies of the stored data will be equally assumed to be exponentially distributed with rate $\kappa > \mu$.

We will focus our analysis on the first year of operation of the small disk array of Fig. 1. We will first consider the case where none of its four disks is more than two-third full and we do not receive any early warning of future disk failures. Fig. 8 displays the state probability transition diagram of that array. State <2, 2> is the normal state of the array when its four disks are operational and each of its two data sets is mirrored on two disks. Observe that all three other states have transitions of
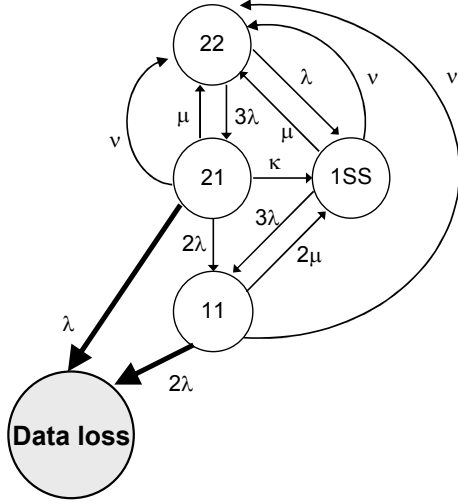
Fig. 9. State transition diagram for the same self-adaptive array of four drives assuming that it receives an early warning of 25 percent of disk failures.

rate ν that return to state $\langle 2, 2\rangle$. They return the array to its normal state after a period of average duration $1/\nu$. Selecting a value of ν equal to one transition per year ensures that we will only consider the behavior of the array during a period whose average duration corresponds to the first year of operation of the array.

When one of its four disks fails the system goes from state $\langle 2, 2\rangle$ to state $\langle 2, 1\rangle$, that is the state of the array depicted in Fig. 2. This state is a less than desirable state as one of the two original data sets has lost one of its two mirrored copies. Hence a failure of the disk containing the remaining copy of that data set would result in a data loss. To avoid that possibility, the array will create an additional copy of that data that will be stored on two of the three remaining disk. This will move the array to state $\langle 1, SS\rangle$ with the letter S denoting a disk has one and half copy of the original data. State $\langle 1, SS\rangle$ is the state of the array depicted in Fig. 3. A failure of any of the three remaining drives will bring the array into state $\langle 1, 1\rangle$ where each of the remaining disks has one complete copy of one of the two data sets. A failure of either of these two disks will therefore result in a data loss.

Repair transitions go from state $\langle 1, 1\rangle$ to state $\langle 1, SS\rangle$ and from both states $\langle 2, 1\rangle$ and $\langle 1, SS\rangle$ to state $\langle 2, 2\rangle$.

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

$$\frac{dp_{22}(t)}{dt} = -4\lambda p_{22}(t) + \mu(p_{21}(t) + p_{1SS}(t)) + \nu(p_{21}(t) + p_{1SS}(t) + p_{11}(t))$$

$$\frac{dp_{21}(t)}{dt} = -(3\lambda + \kappa + \nu)p_{21}(t) + 4\lambda p_{22}(t)$$

$$\frac{dp_{1SS}(t)}{dt} = -(3\lambda + \nu)p_{1SS}(t) + \kappa p_{21}(t) + 2\mu p_{11}(t)$$

$$\frac{dp_{11}(t)}{dt} = -(2\lambda + \nu)p_{11}(t) + \lambda p_{21}(t) + 2\lambda p_{1SS}(t)$$

where $p_{ij}(t)$ is the probability that the system is in state $\langle i, j\rangle$ with the initial conditions $p_{22}(0) = 1$ and $p_{ij}(0) = 0$ for all other states.

The Laplace transforms of these equations are

$$sp_{22}^*(s) - 1 = -4\lambda p_{22}^*(s) + \mu(p_{21}^*(s) + p_{1SS}^*(s)) + \nu(sp_{21}^*(s) + p_{1SS}^*(s) + p_{11}^*(s))$$

$$sp_{21}^*(s) = -(3\lambda + \kappa + \nu)p_{21}^*(s) + 4\lambda p_{22}^*(s)$$

$$sp_{1SS}^*(s) = -(3\lambda + \nu)p_{1SS}^*(s) + \kappa p_{21}^*(s) + 2\mu p_{11}^*(s)$$

$$sp_{11}^*(s) = -(2\lambda + \nu)p_{11}^*(s) + \lambda p_{21}^*(s) + 2\lambda p_{1SS}^*(s)$$

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

$$MTTDL = \sum_i p_i^*(0),$$

we solve the system of Laplace transforms for $s = 0$ and use this result to compute the MTTDL and the mean failure rate (1/MTTDL). The expressions we obtain are quotients of two polynomials that are too large to be displayed.

We have supposed so far that our disk array did not attempt to anticipate disk failures. Since S.M.A.R.T. technology can successfully predict about 30 percent of future failures, we will assume that a disk array starting to reorganize itself whenever it receives a warning of a pending failure will be able to reorganize itself before the failure occurs 25 percent of the times.

Fig. 9 displays the state probability transition diagram for the small array of Fig. 1 assuming that it can now reorganize itself before a failure occurs 25 percent of the times. As we can see, this state probability transition diagram is almost identical to that of Fig. 8: the sole difference between the two diagrams is the transitions leaving state $\langle 2, 2\rangle$. In Fig. 8, a disk failure always brings the array from state $\langle 2, 2\rangle$ to state $\langle 2, 1\rangle$. This is not true in Fig. 9 as a disk failure occurring while the array is in state $\langle 2, 2\rangle$ will bring the array to state $\langle 1, SS\rangle$ 25 percent of the times and to state $\langle 2, 1\rangle$ 75 percent of the time. The first transition corresponds to a failure that was anticipated early enough to complete the reorganization process before the failure occurred while the second transition corresponds to either a sudden failure or a failure that was anticipated too late to complete the restructuring process.

Given the strong similarity, between the two diagrams, we did not feel necessary to give the details of the computation of the mean failure rate of the array.

Fig. 10 displays on a logarithmic scale the probability of a data loss during the first year of the lifetime of the disk array. We assumed that the disk failure rate λ during the first year was one failure every one hundred thousand hours, that is, slightly less than one failure every eleven years. We let the average disk repair times vary between one half-day and one week and considered the two cases where the reorganization process could either take one or four hours.

As we can see, the data loss probabilities achieved by our self-adaptive technique are significantly lower than those achieved by a static array. The best results are obtained for a
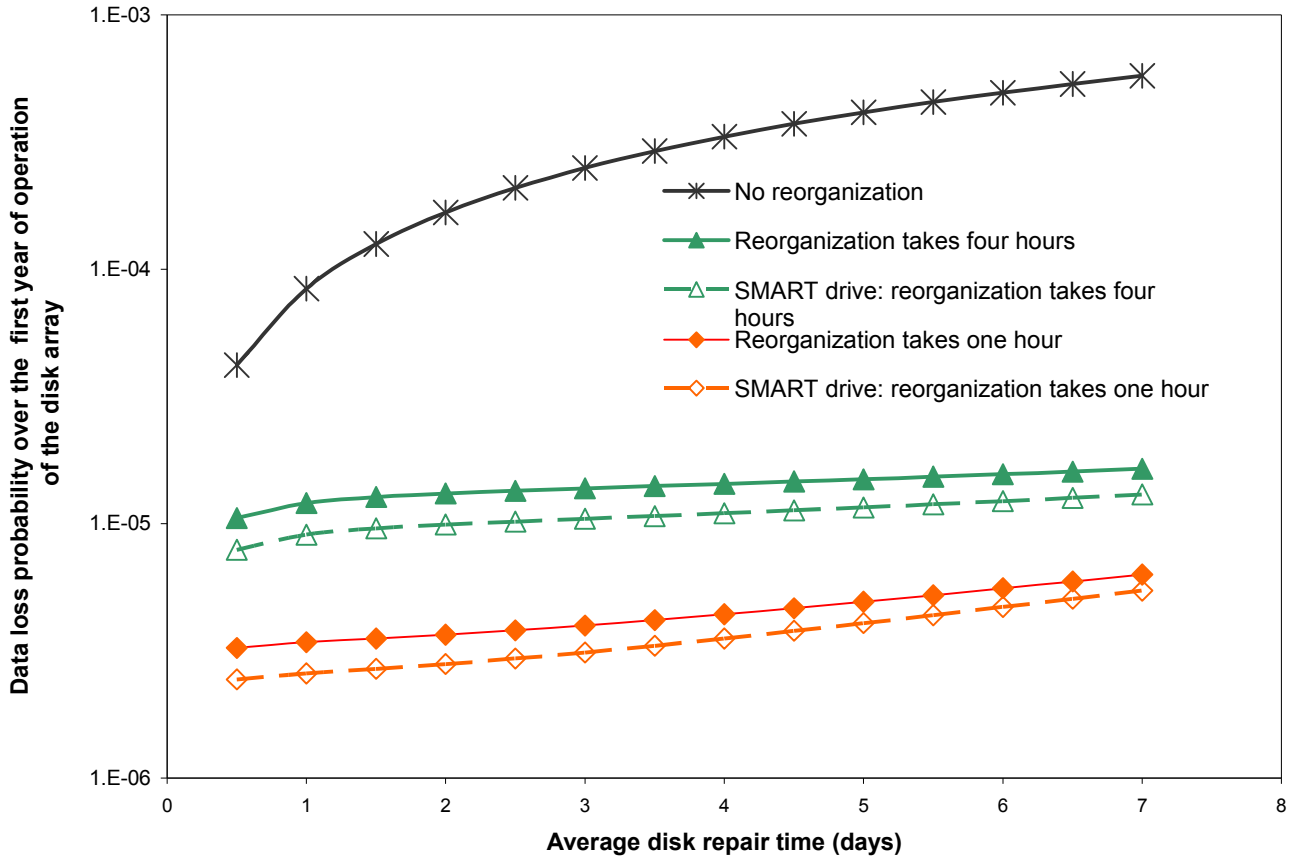
Fig. 9. Array failure rates during its first year of operation assuming that each disk drive is at most 66 percent full.

combination of a fast reorganization process (high κ) and a long repair time (low μ) as the reorganization process keeps the data protected during most of the repair process. Conversely, the reorganization process has much less impact on the array data loss probability when we have both a relatively slow reorganization process and a relatively fast repair process. Even then, the benefits of the reorganization process remain clear: our technique will always reduce the probability of a data loss during the first year of operation of the system by at least 75 percent provided the disk repair process takes at least 12 hours and the reorganization process takes at most 4 hours.

We can also observe the benefits of using S.M.A.R.T. technology to obtain early warnings of future disk failures and to initiate the reorganization process without waiting for the occurrence of the failure. These benefits are fairly limited as we assumed that S.M.A.R.T. technology could only predict 30 percent of disk failures and the array could complete its reorganization before the failure occurred 25 percent of the time. These are fairly conservative assumptions. Hughes *et al.* [4] have claimed that S.M.A.R.T. technology could actually predict between 50 and 60 percent of disk failures. If this was the case, S.M.A.R.T. technology could have a more dramatic impact on effectiveness of our technique.

We can also observe that the failure rates achieved by our self-adaptive array during its first year of operation remain nearly constant over a wide range of disk repair times. This is a significant advantage because fast repair times require maintaining a local pool of spare disks and having maintenance personnel on call 24 hours a day. Since our self-adaptive organization tolerates repair times of up to one week, if not more, it will be cheaper and easier to maintain than a comparable static mirrored disk array with the same number of disks. This will result in a significant decrease of the total cost of ownership of the array.

We also investigated the potential benefits of having more free space on each disk. As we observed in Section II, a mirrored disk array that is only half full can adapt itself to the failures of up to three of its four disks. The behavior of such a disk array could only be modeled by a fairly complex state probability transition diagram that involved too many transitions to be properly displayed. We were expecting that the array would have a much lower risk of data loss than a disk array that is two-thirds full and can only tolerate the failure of two of its four disks. As Fig. 10 indicates, it was not the case: the data loss probabilities of the two arrays were virtually undistinguishable as long as the average disk repair time remained below three to four days. The most likely
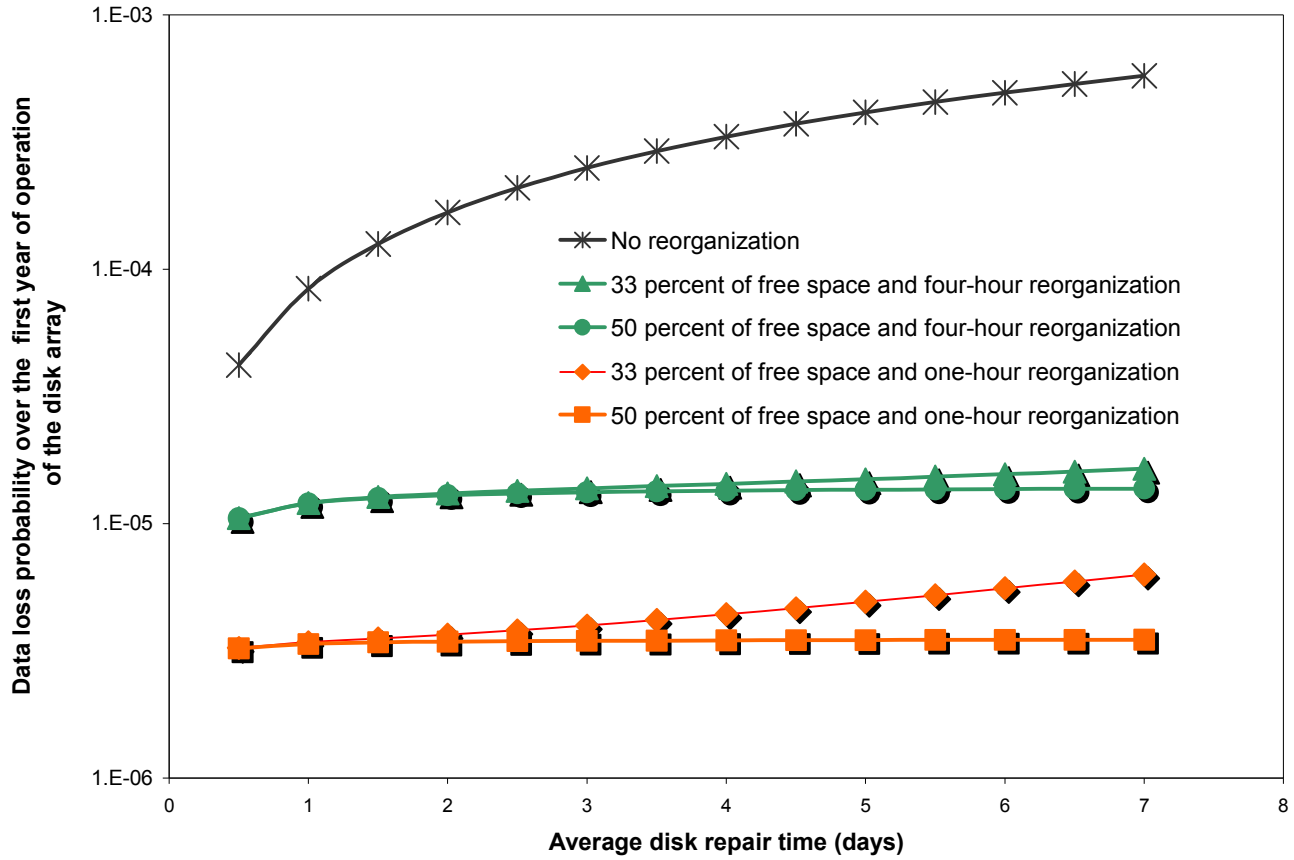
Fig.10. How the percentage of free space affects the failure rate of an array consisting of two pairs of mirrored disks.

explanation is that both arrays are unlikely to experience the failure of more than two of its four disks during such a short interval.

A last issue to consider is the applicability of our technique to larger disk arrays. We have only considered so far a very small array consisting of four disks as increasing its size would have greatly complicated our model. We can reasonably expect our technique to work as well, if not better; in larger disk arrays as these arrays will be more likely to have at least a pair of disks whose spare spaces are not involved in some previous reconfiguration process.

V. CONCLUSIONS

We have presented a new technique for protecting mirrored data against the increased risk of disk failures during their first year of operation. When all disks are operational, all data are mirrored on two disks. Whenever a disk fails, the system reorganizes itself, by storing a new copy of the disk that failed on one or more disks that have free space. This will protect the data against any single disk failure until the failed disk gets replaced and the system reverts to its original state.

To evaluate the benefits of this new disk organization, we have analyzed the behavior of a small system consisting of two pairs of mirrored disks under standard Markovian assumptions. Our results indicate that our technique can reduce the probability of a data loss during the first year of operation of the system by at least 75 percent. We also found out that even better results could be achieved by taking advantage of the failure prediction capabilities of the new S.M.A.R.T. disks [4, 9, 13].

More work is still needed to evaluate the performance of our technique on larger disk arrays, investigate more realistic repair time distributions and measure the impact of our technique on the data survival rate over the whole lifetime of a disk array.

REFERENCES

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

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

[3]   J. G. Elerath. "Specifying reliability in the disk drive industry: No more MTBF's," *Proc. 2000 Annual Reliability and Maintainability Symposium*, pp. 194–199, Jan. 2000,

[4]   G. F. Hughes, J. F. Murray, K. Kreutz-Delgado and C. Elkan. "Improved disk-drive failure warnings," *IEEE Transactions on Reliability*,51(3): 50–357, Sep. 2002.

[5]   P. Lyman and H. R. Varian. "How much information?" *The Journal of Electronic Publishing*, http://www.press.umich.edu/jep, Vol. 6, No. 2, Dec. 2000.

[6]   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,* pp. 109–116, June 1988.

[7]   C. Pu, J. D. Noe and A. Proudfoot. "Regeneration of replicated objects: a technique and its Eden implementation." *IEEE Transactions on Software Engineering,* 14(7): 936–945, Jul. 1988.

[8]   J.-F. Pâris, T. J. E, Schwarz and D. D. E. Long. "Self-adaptive disk arrays," *Proc. 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems* (SSS 2006), Dallas, TX, pp. 469–483, Nov. 2006.

[9]   *Smart Site FAQ* http://smartlinux.sourceforge.net/smart/index.php, accessed Nov. 23, 2006.

[10]   T. J. E. Schwarz and W. A. Burkhard. "RAID organization and performance," *Proc. 12th International Conference on Distributed Computing Systems*, pp. 318–325, June 1992.

[11]   M. Schulze, G. A. Gibson, R. H. Katz and D. A. Patterson. "How reliable is a RAID?" *Proc. Spring COMPCON '89 Conference*, pp. 118–123, Mar. 1989.

[12]   A. Thomasian and J. Menon. "RAID 5 performance with distributed sparing," *IEEE Transactions on Parallel and Distributed Systems*, 8(6):640–657, June 1997.

[13]   *Self-Monitoring, Analysis and Reporting Technology – Wikipedia, the free encyclopedia*, http://en.wikipedia.org/wiki/Self-Monitoring%2C_Analysis_and_Reporting_Technology, accessed Nov. 23 2006.

[14]   J. Wilkes, R. Golding, C. Stealin and T. Sullivan. "The HP AutoRaid hierarchical storage system," *ACM Transactions on Computer Systems*, 14(1): 1–29, Feb. 1996.

[15]   Q. Xin, T. J. E. Schwarz and E. L. Miller, "Disk infant mortality in large storage systems," *Proc. 13th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems* (MASCOTS '05), pp. 125–134, Aug. 2005.