Using Entanglements to Increase the Reliability of Two-Dimensional Square RAID Arrays

Jehan-François Pâris[1], Vero Estrada-Galiñanes[2], Ahmed Amer[3], Carlos Rincón[1,4]
[1]Department of Computer Science, University of Houston, Houston, TX (USA)
[2]Institut d'informatique, Université de Neuchâtel, Neuchâtel, Switzerland
[3]Department of Computer Engineering, Santa Clara University, Santa Clara, CA (USA)
[4]Department of Computer Science, Universidad del Zulia, Maracaibo, Venezuela

Abstract—Two-dimensional square RAID arrays organize their data disks in such a way that each of them belongs to exactly one row parity stripe and one column parity stripe. Even so, they remain vulnerable to the loss of any given data disk and the parity disks of its two stripes. We show how to eliminate all but one of these fatal triple failures by entangling the parity disks of the array, that is, XORing the contents of each parity disk with that of its predecessors. As a result, our new organization reduces the number of fatal triple failures by 96 to 99 percent and the number of fatal quadruple failures by around 85 percent without the need for any additional hardware.

Keywords—storage systems; magnetic disks; system reliability; fault-tolerance.

I. INTRODUCTION

As our societies become more fully digitized, ever increasing resources are dedicated to the online storage of archival data, that is, data that are unlikely to be modified after their creation, that are not frequently accessed and that have lifetimes measured in years, if not in decades. Designing cost-effective online solutions for storing these data remains an important challenge as their hardware requirements differ from those of conventional file systems in several important ways. First, archival data do not make high demands on the storage systems transfer rates. As a result, magnetic disks still maintain a strong cost advantage over solid-state devices. Second, random writes are extremely rare, even when compared with the infrequent read accesses. Finally, archival data stores have much more stringent fault-tolerance requirements than conventional file systems as they have to preserve the integrity of vast collections of data very long periods of time.

Two-dimensional square RAID arrays present a good example of a storage solution that is better suited for archival applications than for conventional file systems. As seen on Fig. 1, they consist of n^2 data disks and $2n$ parity disks organized in such a way that each data disk belongs to two distinct RAID level 4 parity stripes, namely, a row parity stripe and a column parity stripe. For instance, data disk D_{32} belongs to both the row parity stripe that includes parity disk P_3 and the column parity stripe that includes parity disk Q_2. The main advantage of the organization is that it can tolerate all double disk failures and most triple failures without any data loss. As Fig. 2 shows, the sole fatal triple failures are the failure of a data disk and the simultaneous failures of the parity disks of its row parity stripe and its column parity stripe, for a total of n^2 fatal triple failures. Conversely, the organization is not well suited to conventional storage applications as simultaneous updates of data stored on disks in the same parity stripe would all have to update the contents of the same parity disk.

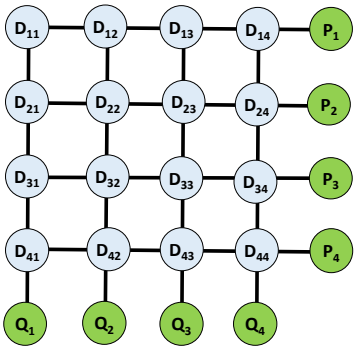

Fig. 1. A two-dimensional square RAID array with 16 data disks and 8 parity disks.

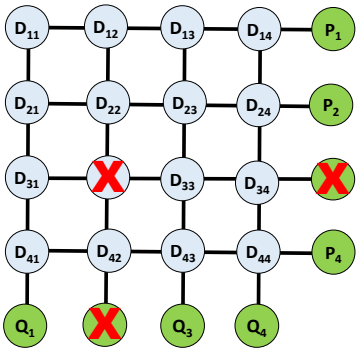

Fig. 2. One of the 16 triple failures that will result in a data loss.

There are however cases when square RAID arrays can fail to provide a sufficient level of protection against data losses. One of such instances is when failed disks cannot be replaced outright. Another is when the disks of the array happen to be much less reliable than expected. As Beach [1] noted, this can happen to disks from the most reputable manufacturers and the failure rates of the bad disks can reach 25 percent per year.

978-1-5090-6468-7/17/$31.00 ©2017 IEEE

The two extant solutions for improving the reliability of square RAID arrays require either adding an additional superparity disk [7] or mirroring either one of the two sets of parity disks [9]. The solution we propose here does not require any additional hardware. It consists of *entangling* each parity disk by XORing its contents with the contents of its immediate predecessor. This will allow the array to recover from all triple disk failures except for the failure of the last data disk and its two parity disks. As a result, the number of fatal triple failures will be reduced by 96 to 99 percent depending on the size of the array. The number of fatal quadruple failures will be similarly reduced by around 85 percent and the number of fatal quintuple failures by 69 to 75 percent.

The remainder of this paper is organized as follows. Section II reviews previous work. Section III introduces our technique and discusses its vulnerability to quadruple and quintuple failures. Section IV evaluates its reliability and compares it to those of conventional two-dimensional RAID arrays. Section V discusses how to implement data updates and Section VI has our conclusions.

II. Previous Work

In this section, we discuss relevant previous work on square RAID arrays and on entanglements.

A. Two-dimensional square arrays

Two-dimensional square RAID arrays, or 2D-parity arrays, were first investigated by Schwarz [11] and by Hellerstein et al. [5]. More recently, Lee patented a two-dimensional disk array organization with prompt parity updates in one dimension and delayed parity updates in the second dimension [6].

Since these arrays store their parity information on dedicated disks, they are better suited for archival storage than maintaining more dynamic workloads.

In a previous paper, some of the authors have proposed to increase the fault-tolerance of these arrays [8] by adding a *superparity* disk S [13] that would contain the exclusive or (XOR) of either all row parity disks, that is, disks P_1 to P_4 in Fig. 1, or all column parity disks, that is, disks Q_1 to Q_4 in the same figure. In other words, we would have

$$S = P_1 \oplus P_2 \oplus P_3 \oplus P_4 = Q_1 \oplus Q_2 \oplus Q_3 \oplus Q_4$$

The extra disk would allow the array to recover from the simultaneous failure of any of the P_i and any of the Q_j parity disks without having to access any data disk, thus eliminating all fatal triple failures.

Another way to eliminate all fatal triple failures [9] is to mirror either all row parity disks (P_1 to P_4 in our example) or all column parity disks (Q_1 to Q_4 in our example).

B. Entanglements

Entanglements [2], [3], [4] trade space for increased reliability and faster writes, especially in the case of log-structured append-only storage systems.

Simple entanglements require equal numbers of data and parity disks; therefore, they have the same space overhead as mirroring. Some of the authors [4] have recently shown that a simple open entanglement chain with $2n$ disks will tolerate the failure of any single disk and the simultaneous failure of any two of them, except for the two last disks, which is much better than a mirrored organization. At the same time, appending a block to the entanglement will require one read and two writes, that is, one less read and one less write than RAID level 6 and two-dimensional RAID arrays.

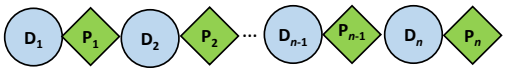

Fig. 3. An open single entanglement.

As Fig. 3 shows, a simple entanglement layout consists of an equal number of data blocks $D_1, D_2, \ldots, D_n$ and parity blocks $P_1, P_2, \ldots, P_n$. As data blocks are added to the entanglement, their associated parity blocks are computed according to the recurrence

$$P_{i+1} = P_i \oplus D_{i+1}$$

with $P_1 = D_1$.

As a result, we have

$$\begin{aligned} P_1 &= D_1 \\ P_2 &= D_1 \oplus D_2 \\ &\ldots \\ P_i &= D_1 \oplus D_2 \ldots \oplus D_{i-1} \oplus D_i \end{aligned}$$

We can eliminate the remaining fatal double failure by *closing* the entanglement and redefining the initial conditions of our recurrence as

$$\begin{aligned} P_1 &= P_i \oplus D_1 \\ P_2 &= D_1 \oplus D_2 \end{aligned}$$

The sole drawback of the process is that appending a new block D_{i+1} will now require updating parity block P_1 in addition to creating a new parity block P_{i+1}.

III. Our Proposal

The best way to increase the reliability of two-dimensional square RAID array is to eliminate as many fatal triple failures as possible. We propose to do that by offering a recovery path for most parity disks involved in a fatal triple failure.

Observe that the parity disks of a square RAID array are defined as

$$\begin{aligned} P_1 &= D_{11} \oplus D_{12} \oplus \ldots \oplus D_{1n}, \\ P_2 &= D_{21} \oplus D_{22} \oplus \ldots \oplus D_{2n}, \\ &\ldots \\ P_n &= D_{n1} \oplus D_{n2} \oplus \ldots \oplus D_{nn} \end{aligned}$$

and

$$\begin{aligned} Q_1 &= D_{11} \oplus D_{21} \oplus \ldots \oplus D_{n1}, \\ Q_2 &= D_{12} \oplus D_{22} \oplus \ldots \oplus D_{n2}, \\ &\ldots \\ Q_n &= D_{1n} \oplus D_{2n} \oplus \ldots \oplus D_{nn}. \end{aligned}$$

We propose a new definition of these parities that entangles:

- Each parity disk P_i with parity disk $P_{i-1}$for $2 \leq i \leq n$
- Each parity disk Q_j with parity disk Q_{n-1} for $2 \leq j \leq n$

As a result,

$$P_1 = D_{11} \oplus D_{12} \oplus \ldots \oplus D_{1n},$$
$$P_2 = P_1 \oplus D_{21} \oplus D_{22} \oplus \ldots \oplus D_{2n},$$
$$\ldots$$
$$P_n = P_{n-1} \oplus D_{n1} \oplus D_{n2} \oplus \ldots \oplus D_{nn}$$

and

$$Q_1 = D_{11} \oplus D_{21} \oplus \ldots \oplus D_{n1},$$
$$Q_2 = Q_1 \oplus D_{12} \oplus D_{22} \oplus \ldots \oplus D_{n2},$$
$$\ldots$$
$$Q_n = Q_{n-1} \oplus D_{1n} \oplus D_{2n} \oplus \ldots \oplus D_{nn}.$$

We now have two ways for recovering the $n-1$ first P_i and Q_j parity disks

$$\begin{aligned} P_1 &= D_{11} \oplus D_{12} \oplus \ldots \oplus D_{1n} \\ &= P_2 \oplus D_{21} \oplus D_{22} \oplus \ldots \oplus D_{2n}, \\ P_2 &= P_1 \oplus D_{21} \oplus D_{22} \oplus \ldots \oplus D_{2n}, \\ &= P_3 \oplus D_{31} \oplus D_{32} \oplus \ldots \oplus D_{3n} \end{aligned}$$
$$\ldots$$
$$\begin{aligned} P_{n-1} &= P_{n-2} \oplus D_{n-1,1} \oplus D_{n-1,2} \oplus \ldots \oplus D_{n-1,n}, \\ &= P_n \oplus D_{n1} \oplus D_{n2} \oplus \ldots \oplus D_{nn}, \end{aligned}$$

and

$$\begin{aligned} Q_1 &= D_{11} \oplus D_{21} \oplus \ldots \oplus D_{n,1} \\ &= Q_2 \oplus D_{12} \oplus D_{22} \oplus \ldots \oplus D_{n,2}, \\ Q_2 &= Q_1 \oplus D_{12} \oplus D_{22} \oplus \ldots \oplus D_{n,2}, \\ &= Q_3 \oplus D_{13} \oplus D_{23} \oplus \ldots \oplus D_{n,3} \end{aligned}$$
$$\ldots$$
$$\begin{aligned} Q_{n-1} &= Q_{n-2} \oplus D_{1,n-1} \oplus D_{2,n-1} \oplus \ldots \oplus D_{n,n-1}, \\ &= Q_n \oplus D_{1n} \oplus D_{2n} \oplus \ldots \oplus_{n,n}. \end{aligned}$$

Consider now the triple disk failure displayed in Fig. 2. It involves data disk D_{32}, row parity disk P_3 and column parity disk Q_2. Entangled parities will offer two ways to recover the lost data:

- Since $P_3 = P_4 \oplus D_{41} \oplus D_{42} \oplus D_{43} \oplus D_{44}$, we now can recover first P_3 then use P_3 to recover D_{32}.
- Since $Q_2 = Q_3 \oplus D_{13} \oplus D_{23} \oplus D_{33} \oplus D_{43}$, we now can recover first Q_2 then use Q_2 to recover D_{32}.

The sole triple failure that would remain irrecoverable will be the simultaneous failure of data disk D_{nn}, row parity disk $P_n$and column parity disk Q_n: as both parity disks $P_n$and Q_n have no successor, we cannot recover them and reconstitute the lost data.

Let us now consider the impact of fatal quadruple and quintuple failures. For convenience, we will use $m = n^2 + 2n$ to denote the total number of disks in the array.

A. Fatal quadruple failures

We can identify four types of fatal quadruple failures:

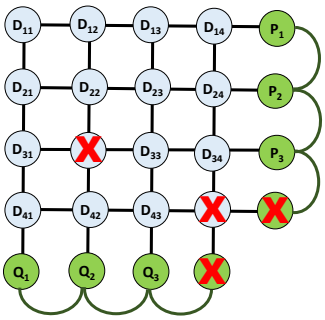

Fig. 4. A type A fatal quadruple failure.

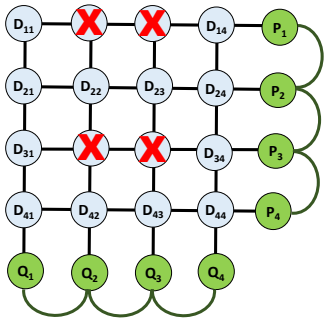

Fig. 5. A type B fatal quadruple failure.

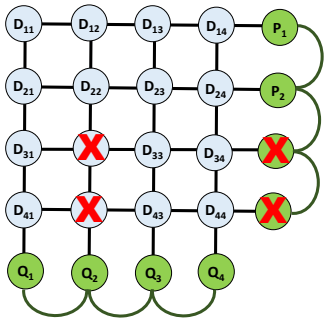

Fig. 6. A type C fatal quadruple failure.

1. As Fig. 4 shows, type A fatal quadruple failures involve the three disks that cause a fatal triple failure plus any one of the remaining disks, for a total of $m - 3$ fatal quadruple failures.
2. As Fig. 5 shows, type B fatal quadruple failures involve the simultaneous failure of four data disks that form a rectangle. There are $\binom{n}{2}^2$ such failures.
3. As Fig. 6 shows, type C fatal quadruple failures involve two data disks and two parity disks that form a rectangle and are all located in either the last two rows or the last two columns of the array. There are $2n$ such failures.
4. There are two type D fatal quadruple failures. Fig. 7 displays the one that involves the last two row parity disks (P_3 and P_4), the last column parity disk (Q_4), and the data disk in the same row as disk P_3 and the same

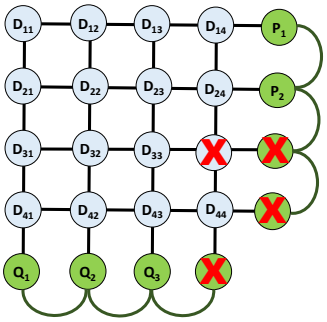

Fig. 7. A type D fatal quadruple failure.

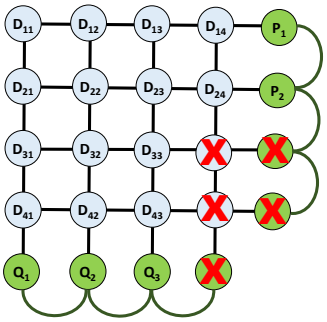

Fig. 8. One of the two fatal quintuple failures that is both as a type A and a type B failure.

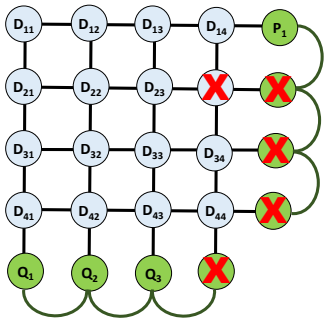

Fig. 9. One of the two type E fatal quintuple failures.

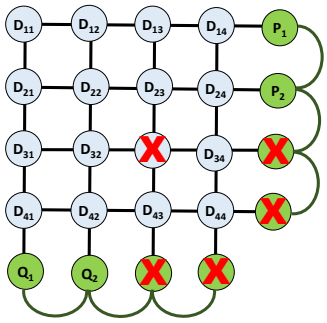

Fig. 10. The sole type F fatal quintuple failure.

column as disk Q_4. The other type D fatal quadruple failure is the symmetric of that one with respect to the main diagonal of the array.

Overall, fatal quadruple failures account for

$$(m-3)+\binom{n}{2}^2+2n\ +\ 2$$

failures out of a total of $\binom{m}{4}$ possible quadruple failures.

B. Fatal quintuple failures

We can distinguish six types of fatal quintuple failures:

1. Fatal quintuple failures of types A to D consist of any group of four disks that would cause a fatal quadruple failure, plus any other disk. There is a total of

$$\binom{m-3}{2}+(\binom{n}{2}^2+2n)(m-4)$$

fatal quintuple failures of types A to D, taking into account that the failure represented in Fig. 8 and its symmetric are both type A as well as type D failures.

2. There are two fatal quintuple failures of type E. One of them is displayed in Fig. 9 and the other is its symmetric with respect to the main diagonal of the array.

3. As Fig. 10 shows, there is a single fatal quintuple failures of type F. It involves the parity disks in the last two rows and two columns and one specific data disk.

Overall, fatal quintuple failures account for

$$\binom{m-3}{2}+(\binom{n}{2}^2+\ 2n)(m-4)+\ 3$$

failures out of a total of $\binom{m}{5}$ possible quintuple failures.

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 intractable as the complexity of the system grows. A simpler option is to use instead the five-year reliability of the array. As this value is typically very close to 1, we will express it in "nines" using the formula $n_n=-log_{10}(1-R_d)$ where R_d is the five-year reliability of the array. Thus, a reliability of 99.9 percent would be represented by three nines, a reliability of 99.99 percent by four nines, and so on.

We develop first a generic Markov model that will apply to all sorts of fault-tolerant storage arrays. The specific behavior of each fault-tolerant disk array will be represented by the four parameters m, α, β, γ and δ where m is the total number of disks in the array and α, β, γ, and δ are the respective probabilities that the array will not tolerate the simultaneous failures of two, three, four or five disks. We will neglect the probability that the array will tolerate a sextuple disk failure, assuming that this probability is small enough to be neglected.

Our model consists of an array of disks with independent failure modes. Whenever a disk fails, a repair process is immediately initiated for that disk. Should several disks fail at

the same time, this repair process will be performed in parallel on those disks. We assume that disk failures are independent events and are exponentially distributed with mean λ. In addition, we require repairs to be exponentially distributed with mean μ. Both hypotheses are necessary to represent our system by a Markov process with a finite number of states.

Fig. 11 displays our state transition probability diagram. State <0> is the initial state where all m disks are operational and no disk has failed. Should any of the disks fail, the system would move to state <1> with an aggregate failure rate $m\lambda$. Whenever fatal double failures are possible, the failure transitions from state <1> will be:

1. A transition to the data loss state with rate $\alpha(m-1)\lambda$ where the actual value of α depends on the specific storage organization.
2. A transition to state <2> with rate $(1-\alpha)(m-1)\lambda$.

In the same way, the two failure transitions from state <2> consist of:

1. A transition to the data loss state with rate $\beta(m-2)\lambda$ where the actual value of β depends on the specific storage organization.
2. A transition to state <3> with rate $(1-\beta)(m-2)\lambda$.

Following the same pattern, the two failure transitions from state <3> are:

1. A transition to the data loss state with rate $\gamma(m-3)\lambda$ where the actual value of γ depends on the specific storage organization.
2. A transition to state <4> with rate $(1-\gamma)(m-3)\lambda$.

In the same way, the two failure transitions from state <4> are:

1. A transition to the data loss state with rate $\delta(m-4)\lambda$ where the actual value of δ depends on the specific storage organization.
2. A transition to state <5> with rate $(1-\delta)(m-4)\lambda$.

Finally, the sole failure transition from state <5> is a transition to the data loss state with rate $(m-5)\lambda$ reflecting our assumption that all sextuple disk failures are fatal.

Recovery transitions are more straightforward: they bring the array from state <5> to state <4>, then from state <4> to state <3> and so on until the system returns to its initial state <0>.

The Kolmogorov system of differential equations that describes the behavior of the array is:

$$\frac{dp_0(t)}{dt} = -m\lambda p_0(t) + \mu p_1(t)$$

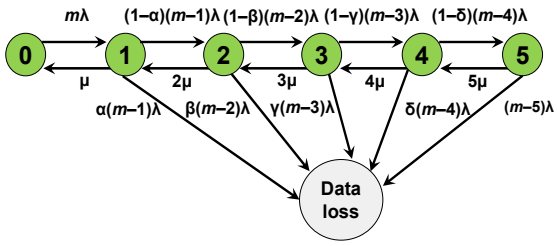

Fig. 11. Our state probability transition diagram.

$$\frac{dp_1(t)}{dt} = -((m-1)\lambda+\mu)p_1(t) + m\lambda p_0(t) + 2\mu p_2(t)$$

$$\frac{dp_2(t)}{dt} = -((m-2)\lambda+2\mu)p_2(t) + \alpha(m-1)\lambda p_1(t) + 3\mu p_3(t)$$

$$\frac{dp_3(t)}{dt} = -((m-3)\lambda+3\mu)p_3(t) + \beta(m-2)\lambda p_2(t) + 4\mu p_4(t)$$

$$\frac{dp_4(t)}{dt} = -((m-4)\lambda+4\mu)p_4(t) + \gamma(m-3)\lambda p_4(t) + 5\mu p_5(t)$$

$$\frac{dp_5(t)}{dt} = -((m-5)\lambda+5\mu)p_5(t) + \delta(m-4)\lambda p_4(t)$$

where $p_i(t)$ is the probability that the system is in state <i> with initial conditions $p_0(0) = 1$ and $p_i(0) = 0$ for $i \neq 0$.

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

$$MTTDL = \sum_{i=0}^{4} p_i^*(\mathrm{s}=0),$$

where $p_i^*(s)$ is the Laplace transform of $p_i(t)$, we compute the Laplace transforms of the above equations and we solve them for $s = 0$ and a fixed value of m [10]. We then use this result to compute the mean time to data loss (MTTDL) of our system and convert this MTTDL into a five-year reliability, using the formula:

$$R_d = \exp\left(-\frac{d}{\mathrm{MTTDL}}\right)$$

where d is a five-year interval expressed in the same units as the MTTDL. Observe that the above formula implicitly assumes that long-term failure rate 1/MTTDL does not significantly differ from the average failure rate over the first five years of the array.

We computed the five-year reliabilities of entangled two-dimensional RAID arrays for three array configurations:

1. A small array with 25 data disks and 10 parity disks.

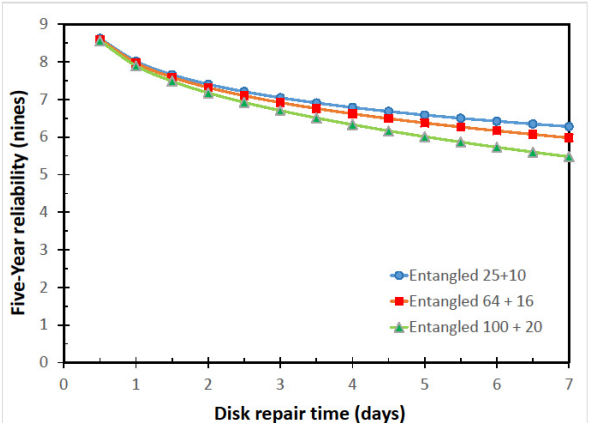

Fig. 12. Five-year reliabilities of the three entangled disk arrays when MTTF = 200,000 hours.

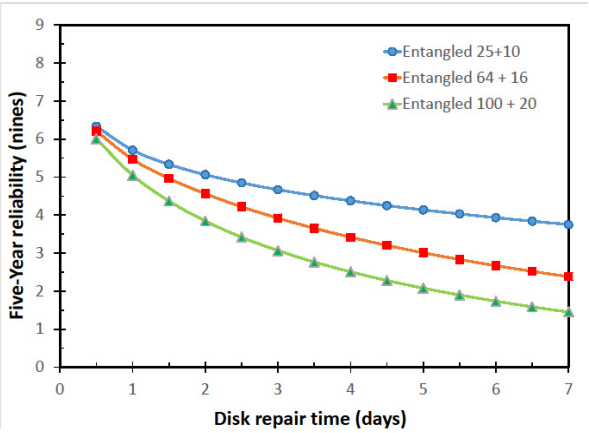

Fig. 13. Five-year reliabilities of the three entangled disk arrays when MTTF =35,000 hours.

2. A medium-size array with 64 data disks and 16 parity disks.
3. A larger array with 100 data disks and 20 parity disks.

and two disk failure rates:

1. A disk MTTF of 200,000 hours, which corresponds to a yearly disk failure rate of 4.28 percent and represents what can be expected from an array built with good disks.
2. A disk MTTF of 35,000 hours, which corresponds to a yearly disk failure rate of 25 percent. While this failure rate is abnormal, it is neither exceptional nor confined to disks of dubious origin [1].

The parameters of our model are:

- m = 35, 80 and 120.
- $\alpha = 0$,
- $\beta = \frac{1}{\binom{m}{3}}$,
- $\gamma = \frac{(m-3)+\binom{n}{2}^2+2n+2}{\binom{m}{4}}$,

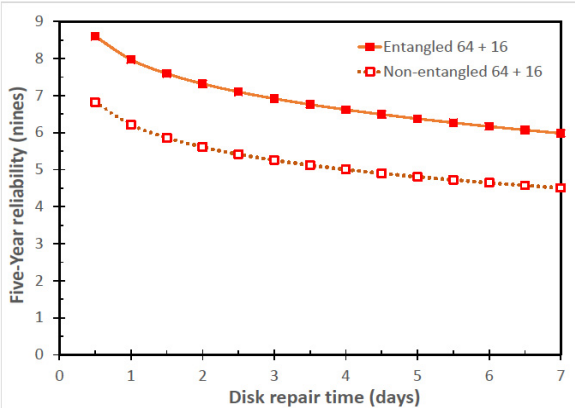

Fig. 14. Compared five-year reliabilities of entangled and non-entangled disk arrays when MTTF = 200,000 hours.

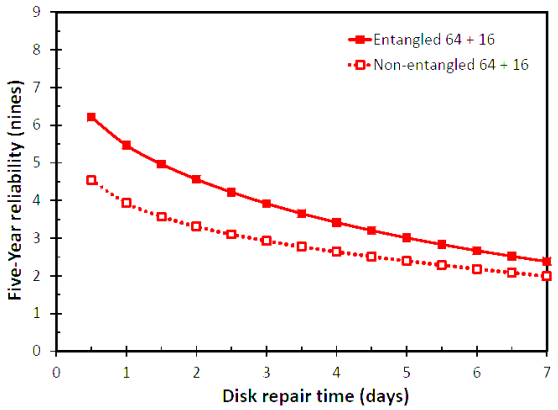

Fig. 15. Compared five-year reliabilities of entangled and non-entangled disk arrays when MTTF = 35,000 hours.

- $\delta = \frac{\binom{m-3}{2}+(\binom{n}{2}^2+2n)(m-4)+3}{\binom{m}{5}}$.

Fig. 12 displays the five-year reliabilities of the three entangled array configurations for a disk MTTF of 200,000 hours and average disk repair times varying between 12 hours and seven days. As we can see, the three configurations provide a six nine (99.9999 percent) reliability over five years as long as the disk repair times do not exceed five days. The same is not true when we consider the case when all the disk arrays belong to a bad batch and have a MTTF of only 35,000 hours. Achieving a five nine (99.999 percent) reliability now requires failing disks to be replaced within 24 hours. The smallest of the three arrays remains significantly more reliable than the two others in part due to its higher parity disk-to-data disk ratio.

A. Comparison with non-entangled 2D RAID arrays

We also compared these results with those achieved by conventional non-entangled two-dimensional RAID arrays. Due to space considerations, we only report here the results for arrays consisting of 64 data disks and 16 parity disks. Results for the two other arrays we investigated are similar.

The parameters of our model for the non-entangled array were [9]:

- $m = 35$,
- $\alpha = 0$,
- $\beta = \frac{n^2}{\binom{m}{3}}$,
- $\gamma = \frac{n^2(m-3) + \binom{n}{2}^2 + 2n\binom{n}{2}}{\binom{m}{4}}$,
- $\delta = \frac{n^2\binom{m-3}{2} + (\binom{n}{2}^2 + 2n\binom{n}{2})(m-4)}{\binom{m}{5}}$.

As Fig. 14 shows, the array with entangled parities performs much better than the conventional non-entangled array under normal operating circumstances, with a disk MTTF of 200,000 hours. In fact, it reduces by 95 to 98 percent the probability of a data loss over a five-year interval. This beneficial effect persists when all the disk arrays belong to a bad batch and have a MTTF of only 35,000 hours, but it tends to decline when the disk repair time exceed 48 hours.

Another, more direct, way to compare the two organizations is to look at their respective numbers of fatal triple, quadruple and quintuple fatal failures. While a conventional 2D square array with $n^2 + 2n$ disks incurs n^2 fatal triple failures, a new entangled organization with the same number of disks eliminates all but one of these failures. For the array sizes we considered, this corresponds to an effective reduction of the number of fatal triple failures from 96 percent for $n = 5$ to 99 percent for $n = 10$. In the same way, the number of fatal quadruple failures will be reduced by around 85 percent and the number of fatal quintuple failures by 69 to 75 percent.

B. *Comparison with simple entanglements*

Our new organization offers two significant advantages over simple entanglements. First it has a much lower space overhead. While single entanglements require equal numbers of data disks and parity disks, entangled 2D arrays only require $2n$ parity disks to protect the contents of n^2 data disks. As a result, an entangled 2D array with 100 data disks and 20 parity disks can store as much data as a simple entanglement with 200 disks.

In addition, entangled 2D arrays protect their data against more multiple failures than simple entanglements. While single entanglements can tolerate, without data loss, *almost all* double failures and most triple, quadruple or quintuple failures, entangled 2D arrays will tolerate, without data loss, *all* double failures, *almost all* triple failures and higher numbers of quadruple or quintuple failures.

It is therefore fair to say that entangled 2D arrays are both cheaper to run and more reliable than simple entanglements.

V. Array Update Issues

Entangled two-dimensional square RAID arrays are more resilient than their non-entangled counterparts because they offer an additional way to recover the contents of failed parity disks. The sole drawback of the approach is that the contents of each parity disk now depends on the contents of all its predecessors, which greatly complicates updates. Consider for instance an entangled array with n^2 data disks, n row parity disks and n column parity disks. Updating the contents of data disk D_{ij} will require updating the contents of row parity disks P_i to P_n and column parity disks Q_j to Q_n. As a result, the average cost of any data update will be $(n+1)/2$ row parity updates and $(n+1)/2$ column parity updates with each parity update involving one read and one write.

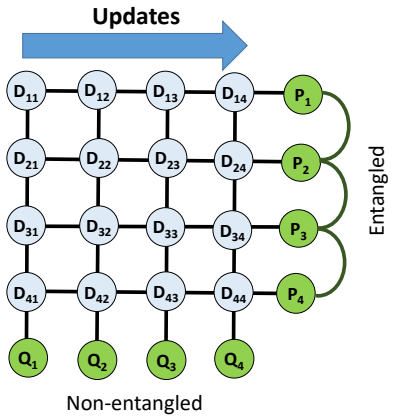

Fig. 16. How partial entangling works.

This high cost might be tolerated during the later phases of the archived data when updates will be the exception. One may even argue that making these updates costlier will help discourage possible tampering. As this not the case when the data are accumulated, we propose two possible options:

1. Whenever random updates cannot be excluded, the best solution is to defer entangling until the archive is not likely to be updated.
2. If the storage array is accessed in append-only mode, we can use partial entanglements. Fig.16 shows an instance of such an organization. All writes are performed sequentially in row major order and never proceed to the next row until the current row is full. As a result, the row parity disks can be safely entangled as all row parity disks are updated in sequence. The sole drawback of the solution is the lesser protection it affords because the array will not be able to tolerate any triple failure involving one of the n last data disks of the log and its respective row and column parity disks.

VI. Conclusion

We have presented an entangled organization for two-dimensional RAID arrays that greatly increases their reliability without requiring any additional hardware. Since our new organization makes the content each respective row and column parity disk dependent on the contents of all its predecessors, it provides an additional way to recover the contents of failed parity disks, which eliminates all but one fatal triple failures. As a result, our new organization minimizes the number of fatal triple failures by 96 to 99 percent, the number of quadruple failures by approximately 85 percent, and the number of quintuple failures by 69 to 75 percent, when compared against a conventional 2D square array.

A stochastic analysis of the new organization has shown that it can provide six nine (99.9999 percent) reliability over five years as long as (a) the disk MTTF does not fall below 200,000 hours, (b) disk repair times do not exceed five days, and (c) the disk array size does not exceed 120 disks.

More work is still needed to evaluate the impact of declustering strategies such as disklets to the reliability of entangled arrays [12].

REFERENCES

[1] B. Beach, "What hard drive should I buy?" https://www.backblaze.com/blog/what-hard-drive-should-i-buy/, retrieved Oct. 20, 2017.

[2] V. Estrada and P. Felber, 'Helical entanglement codes: An efficient approach for designing robust distributed storage systems," Proc. 15th Int. Symp. on Stabilization, Safety, and Security of Distributed Systems, pp 32-44, Nov. 2013.

[3] V. Estrada and P. Felber, "Ensuring data durability with increasingly interdependent content," Proc. IEEE 2015 Int. Conf. on Cluster Computing, pp. 162–165, Sep. 2015.

[4] V. Estrada-Galiñanes, J.-F. Pâris and P. Felber, "Simple data entanglement layouts with high reliability," Proc. 35th Int. Performance of Computers and Communication Conf., Dec. 2016.

[5] L. Hellerstein, G. Gibson, R. M. Karp, R. H. Katz, and D.A. Patterson, " Coding techniques for handling failures in large disk arrays." Algorithmica, 12(3-4):182-208, June 1994

[6] W. S. Lee, "Two-dimensional storage array with prompt parity in one dimension and delayed parity in a second dimension," US Patent #6675318 B1, 2004

[7] J.-F. Pâris and A. Amer. "Using shared parity disks to improve the reliability of RAID arrays, Proc. 28th Int. Performance of Computers and Communication Conf., pp. 129–136, Dec. 2009.

[8] J.-F. Pâris, T. Schwarz, S. J., A. Amer and D. D. E. Long, "Highly Reliable Two-Dimensional RAID Arrays for Archival Storage," Proc. 31st Int. Performance of Computers and Communication Conf., pp. 324–331, Dec. 2012

[9] J.-F. Pâris, T. Schwarz, S. J., A. Amer and D. D. E. Long, "Protecting RAID Arrays Against Unexpectedly High Disk Failure Rates," Proc. 20th IEEE Pacific Rim Int. Symp. on Dependable Computing, pp. 68–75, Nov. 2014.

[10] M. Rausand, A. Høyland, System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Edition, Wiley, 2003.

[11] T. Schwarz, S. J., Reliability and Performance of Disk Arrays, PhD Dissertation, Department of Computer Science and Engineering, University of California, San Diego, 1994.

[12] T. Schwarz, S. J., A. Amer, T. Kroeger, E. L. Miller, D. D. E. Long and J.-F. Pâris, "Reliable Storage at Exabyte Scale," Proc. 24th Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Sep. 2016.

[13] A. Wildani, T. Schwarz, S. J., E. L. Miller and D. D. E. Long, "Protecting against rare event failures in archival systems," Proc. 17th IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sep. 2009.