Protecting RAID Arrays Against Unexpectedly High Disk Failure Rates

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

Thomas Schwarz, S. J.
Depto. de ICC
Universidad Católica del Uruguay
Montevideo, Uruguay
tschwarz@ucu.edu.uy

Ahmed Amer
Computer Eng. Dept.
Santa Clara University
Santa Clara, CA
aamer@scu.edu

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

***Abstract*—Disk failure rates vary so widely among different makes and models that designing storage solutions for the worst case scenario is a losing proposition. The approach we propose here is to design our storage solutions for the most probable case while incorporating in our design the option of adding extra redundancy when we find out that its disks are less reliable than expected. To illustrate our proposal, we show how to increase the reliability of existing two-dimensional disk arrays with n^2 data elements and $2n$ parity elements by adding n additional parity elements that will mirror the contents of half the existing parity elements. Our approach offers the three advantages of being easy to deploy, not affecting the complexity of parity calculations, and providing a five-year reliability of 99.999 percent in the face of catastrophic levels of data loss where the array would lose up to a quarter of its storage capacity in a year.**

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

I. INTRODUCTION

As we produce ever increasing amounts of new data year after year, we are faced with the problem of preserving data over lifetimes that can extend over several decades. For instance, most people expect their family pictures to remain available for their children and grandchildren.

Storing digital data online can offer the two advantages of making them instantly accessible and protecting them against medium obsolescence or degradation. The only drawback of the approach is the need to protect such data against human error and equipment failures. This requirement has led to various redundancy schemes, among which are: mirroring, triplication and various RAID organizations.

Finding the best redundancy scheme for a given application assumes that we know the reliability requirements of the application, its expected data access patterns, and the likelihood of equipment failures over the lifetime of the data. Two large experimental studies of disk reliability [16, 19] showed that disk failure rates typically remain below 8 to 9 percent per year. This meant that a disk mean time to failure (MTTF) of 100,000 hours was a good conservative estimate of disk reliability. A more recent study [1] challenges this assumption by showing that these average values mask important differences among disk makes and models. While its author reported that disks from the best makes and models were found to have yearly failure rates between 2 and 4 percent, he also reported a 25 percent failure rate for a batch of 539 disks coming from a reputable manufacturer.

These new data show that architects of storage solutions cannot design their disk arrays for an average disk failure rate of, say, 4 to 8 percent per year without running the risk of not meeting the data reliability requirements of their applications. At the same time, assuming the worst case scenario would result in unnecessarily complex array organizations that would overprotect their data.

We propose a new variable-redundancy approach for handling this uncertainty. It consists of designing storage solutions for the most probable case while incorporating in our design the possibility of quickly increasing the redundancy level of the array if its components prove themselves to be less reliable than expected. For instance, we could move from an array organization that withstands double disk failures to one that tolerates triple failures. In the same way, we could introduce intra-disk parity in our array if irrecoverable read errors are the problem. We call this process *array hardening*.

To illustrate our proposal, we show how it applies to two-dimensional square arrays with n^2 data disks and $2n$ parity disks. Such disk arrays protect stored data against all double as well as most triple and quadruple disk failures. Should their component disks prove themselves less reliable than expected, protecting the array against all triple disk failures may become necessary. To achieve this goal, we can add to the array n additional parity disks that will mirror the contents of one half of the existing parity disks. The new array will have a total of $n^2 + 3n$ disks and will be able to tolerate all triple failures. To give an example of the effectiveness of this approach, let us consider a disk array with 64 data disks and 16 parity disks. Data will remain well protected as long as the disk failure rate remains around 4 percent per year. When this is not the case, it becomes increasingly difficult to guarantee long-term survival of the data. Adding eight extra parity disks allows the array to achieve a five-year data survival rate of 99.999 percent when the disk failure rate is 25 percent per year as long as failed disks are replaced within 36 hours.

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

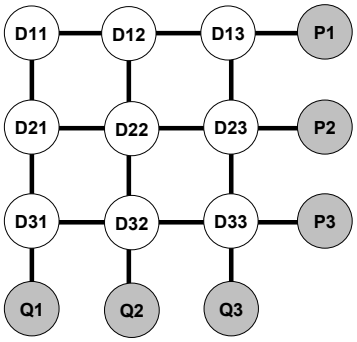

Fig. 1. A two-dimensional RAID array with 9 data disks and 6 parity disks.

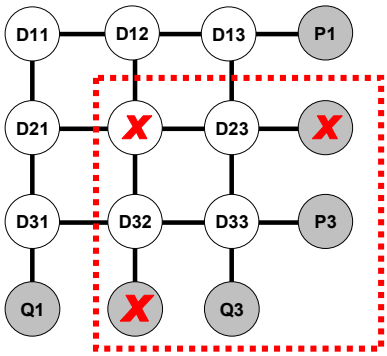

Fig. 2. A sample triple failure resulting in a data loss.

and compares it to those of conventional two-dimensional RAID arrays and two-dimensional RAID arrays comprising a superparity device. Section V discusses possible extensions and Section VI has our conclusions.

II. Previous Work

RAID arrays were the first disk array organizations to utilize erasure coding in order to protect data against disk failures [5, 15, 21]. While RAID levels 3, 4 and 5 only tolerate single disk failures, RAID level 6 organizations use $(n-2)$-out-of-n codes to protect data against double disk failures [3]. EvenOdd, Row-Diagonal Parity and the Liberation Codes use only XOR operations to construct their parity information [2, 4, 6, 17, 18]. Huang and Xu proposed a coding scheme correcting triple failures [8].

Two-dimensional RAID arrays, or 2D-Parity arrays, were first investigated by Schwarz [20] and by Hellerstein et al. [7]. 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 [9]. Since these arrays store their parity information on dedicated disks, they are better suited for archival storage than maintaining more dynamic workloads.

We will focus this investigation on two-dimensional RAID arrays, or 2D-Parity arrays, with n^2 data disks and $2n$ parity disks, such as the one represented in Fig. 1. Each of its nine data disks belongs to both a row-wise parity stripe and a column-wise parity stripe. For instance, data disk D_{22}

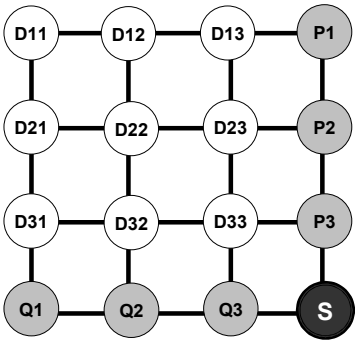

Fig. 3. A two-dimensional RAID array with 9 data disks, 6 parity disks and one superparity device (S).

belongs to both parity stripe (D_{21}, D_{22}, D_{23}, P_2), whose parity disk is disk P_2, and parity stripe (D_{12}, D_{22}, D_{32}, Q_2), whose parity disk is disk Q_2. Pâris et al. [11] noted that these arrays tolerated all double and all triple failures but the failures of a data disk D_{ij} and its two parity disks P_i and Q_j. Fig. 2 displays one of these fatal triple failures, which involves data disk D_{22} and parity disks P_2 and Q_2.

A simple way to increase the fault-tolerance of these arrays [14] is to add a *superparity* device S [23] that contains the exclusive or (XOR) of either all row parity disks, that is, disks P_1 to P_3 in Fig. 1, or all column parity disks, that is, disks Q_1 to Q_3 in the same figure. In other words we would have

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

Fig. 3 illustrates one such organization. Observe that the array can now use its superparity device S 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. The main disadvantage of this solution is that all data updates must now be propagated to the superparity device, which drastically reduces the array update rate. As a result, this solution only applies to mature archival stores that will be infrequently updated.

III. Our Approach

We want to address the issue of the wide variability of disk failure rates, even among disks coming from reputable manufacturers. Beach [1], surveying a population of 27,134 consumer-grade drives spinning at Backblaze, reports typical disk failure rates of less then 5 percent per year but mentions failure rates as high as 25 percent per year for a batch of 539 1.5TB disks coming from a reputable manufacturer. This uncertainty puts storage array architects in a conundrum. Should they design their arrays assuming a disk failure rate of 4 to 5 percent per year or take into account the possibility of much higher disk failure rates? In the first case, they would run the risk of losing data if disks prove themselves to be less reliable than expected. Adding enough redundancy to tolerate a much higher rate is not a much more attractive proposition, as it would result in

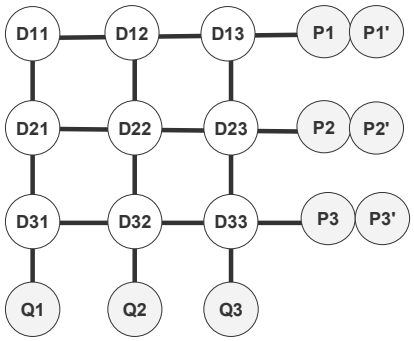

Fig. 4. A two-dimensional RAID array with 9 data disks, 3 mirrored row parity disks and three column parity disks.

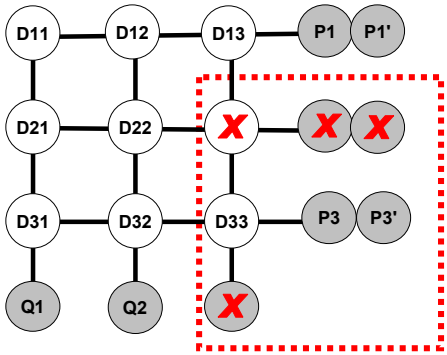

Fig. 5. A sample Type A fatal quadruple failure consisting of the failure of one data disk and its three parity disks.

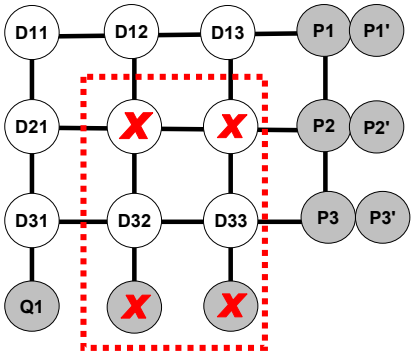

Fig. 6. A sample Type B fatal quadruple failure consisting of the failure of two data disks in the same row and their two column parity disks.

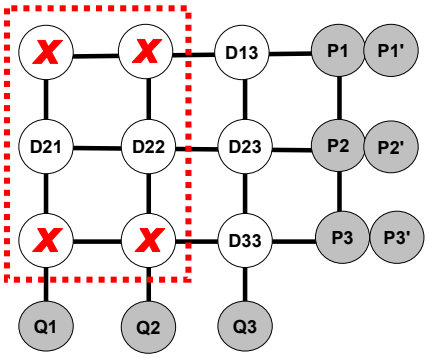

Fig. 7. A sample Type C fatal quadruple failure consisting of the failure of two data disks in the same row and their two column parity disks.

costlier arrays, with more complicated update procedures and higher operating costs.

We propose an adaptive approach that will let the array architecture adapt to observed disk failure rate increases by improving its fault-tolerance. This would allow us to design the array for the most likely scenario where disk failure rates do not exceed 4 to 5 percent. Should observed failure rates become much higher than that, we would add extra redundancy to the array in order to make it more fault-tolerant. The main advantage of this approach is that we will only pay for this extra redundancy when needed.

Two-dimensional square arrays with n^2 data disks and $2n$ parity disks provide a good case study for this new approach. Under normal conditions, these arrays provide more than adequate protection for most archival storage applications. The main exception is when the array experiences higher than expected failure rates due either to environmental conditions, such as vibrations or poor cooling conditions, or to a bad batch of disks. When this is the case, we cannot exclude the risk of a triple failure resulting in a data loss.

To eliminate this risk, we must eliminate all fatal triple failures. We want our solution to satisfy two important conditions. First, the upgrade should be easy to apply and should not require a complete reorganization of the array. Our objective is to increase the fault-tolerance of an existing array rather than building a new one. Second, the upgrade should have a minimum impact on the update bandwidth of the array. This second requirement excludes the use of a single superparity device.

Returning to our original problem, we observe that all fatal triple failures involve the failure of a data disk and its two parity disks. Thus mirroring either *all row parity disks* (disks P_1 to P_3 in our examples) or *all column parity disks* (disks Q_1 to Q_3 in our examples) would suffice to eliminate all fatal triple failures. Fig. 4 displays an instance of this new organization where the row parity disks P_1 to P_3 are mirrored. Observe that the three column parity disks Q_1 to Q_3 are left unchanged: mirroring them is not needed in order to eliminate all fatal triple failures and would not suffice to eliminate all quadruple fatal failures. We call this technique *on-demand partial parity mirroring*.

A main advantage of our technique is its low impact on update procedures. Since the same parity information will be used to update the original row parity disks P_i and their mirrors P_i', our organization does not require any additional parity calculations. Installing the upgrade will not require putting the whole array offline and can be performed row by row with each step resulting in the elimination of some fatal triple failures.

Let us now consider which types of quadruple failures will remain fatal. These fatal failures include:

1. The failure of a data disk and its three parity disks: Fig. 5 shows one of these type A failures.
2. The failure of two data disks in the same row and their two column parity disks: Fig. 6 shows one of these type B failures.
3. The failure of four data disks forming a rectangle: Fig. 7 shows one of these type C failures.

Consider now an array with n^2 data disks and $2n$ parity disks, n of which are mirrored, for a total of $n^2 + 3n$ disks. Out of the $\binom{n^2+3n}{4}$ possible quadruple failures the array can experience, we can enumerate:

1. n^2 type A failures,
2. $n\binom{n}{2}$ type B failures,
3. $\binom{n}{2}^2$ type C failures,

for a total of

$$n^2 + n\binom{n}{2} + \binom{n}{2}^2 = \frac{n^4 + 3n^2}{4}$$

fatal quadruple failures.

As the size of the array grows, the ratio between the number of fatal quadruple failures and the total number of quadruple failures

$$\frac{(n^4+3n^2)/4}{\binom{n^2+3n}{4}}$$

will quickly decrease: it becomes less than 0.4 percent for $n \geq 4$ and less than 0.05 percent for $n \geq 8$.

Let us now turn our attention to quintuple failures and enumerate which ones will result in a data loss. These fatal failures will consist of all quadruple fatal failures plus any other disk. Out of the $\binom{n^2+3n}{5}$ possible quintuple failures the array can experience, we can thus enumerate

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

quintuple failures leading to a data loss.

As we noted before for fatal quadruple failures, the fraction of fatal quintuple failures also decreases with the size of the array; it becomes less than 2 percent for $n \geq 4$ and less than 0.3 percent for $n \geq 8$.

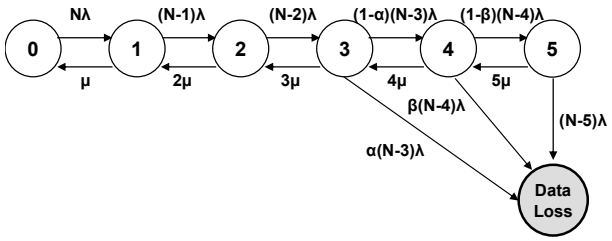

Fig. 8. State transition probability diagram for a two-dimensional RAID array with N = $n^2 + 3n$ disks.

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.

Our system 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, 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. 8 displays the state transition probability diagram for a two-dimensional RAID array with n^2 data disks and $3n$ parity disks for a total of $N = n^2 + 3n$ disks. State <0> is the original state where all N disks are operational and no disk has failed. Should one of the disks fail, the system would move to state <1> with an aggregate failure rate $N\lambda$. A second failure would bring the system to state <2>. A third failure would bring the system to state <3>. Since some quadruple failures are fatal, the two failure transitions from state <3> will be:

1. A transition to the failure state with rate $\alpha(N-3)\lambda$ where
$$\alpha = \frac{(n^4+3n^2)/4}{\binom{n^2+3n}{4}}$$
2. A transition to state <4> with rate $(1-\alpha)(N-3)\lambda$.

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

1. A transition to the failure state with rate $\beta(N-3)\lambda$ where

$$\beta = \frac{(n^4+3n^2)(n^2+3n-4)/4}{\binom{n^2+3n}{5}}$$

2. A transition to state <5> with rate $(1-\beta)(N-3)\lambda$.

As we did not take into account the possibility that the array could survive a sextuple failure, there is a single failure transition leaving state <5>.

Recovery transitions are more straightforward: they bring the array from state <4> to state <3>, then from state <3> to state <2> and so on until the system returns to its original state <0>.

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

$$\begin{aligned}
\frac{dp_0(t)}{dt} &= -N\lambda p_0(t) + \mu p_1(t) \\
\frac{dp_1(t)}{dt} &= -((N-1)\lambda+\mu)p_1(t) + N\lambda p_0(t) + 2\mu p_2(t) \\
\frac{dp_2(t)}{dt} &= -((N-2)\lambda+2\mu)p_2(t) + (N-1)\lambda p_1(t) + \\
&\quad 3\mu p_3(t) \\
\frac{dp_3(t)}{dt} &= -((N-3)\lambda+3\mu)p_3(t) + (N-2)\lambda p_2(t) + \\
&\quad 4\mu p_4(t) \\
\frac{dp_4(t)}{dt} &= -((N-4)\lambda+4\mu)p_4(t) + (1-\alpha)(N-3)\lambda p_3(t) + \\
&\quad 5\mu p_5(t) \\
\frac{dp_5(t)}{dt} &= -((N-5)\lambda+5\mu)p_5(t) + (1-\beta)(N-4)\lambda p_4(t)
\end{aligned}$$

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

The Laplace transforms of these equations are

$$\begin{aligned}
sp_0^*(s) - 1 &= -N\lambda p_0^*(s) + \mu p_1^*(s) \\
sp_1^*(s) &= -((N-1)\lambda+\mu)p_1^*(s) + N\lambda p_0^*(s) + 2\mu p_2^*(s) \\
sp_2^*(s) &= -((N-2)\lambda+2\mu)p_2^*(s) + (N-1)\lambda p_1^*(s) + \\
&\quad 3\mu p_3^*(s) \\
sp_3^*(s) &= -((N-3)\lambda+3\mu)p_3^*(s) + (N-2)\lambda p_2^*(s) + \\
&\quad 4\mu p_4^*(s) \\
sp_4^*(s) &= -((N-4)\lambda+4\mu)p_4^*(s) + \\
&\quad (1-\alpha)(N-3)\lambda p_3^*(s) + 5\mu p_5^*(s) \\
sp_5^*(s) &= -((N-5)\lambda+5\mu)p_5^*(s) + \\
&\quad (1-\beta)(N-4)\lambda p_4^*(s)
\end{aligned}$$

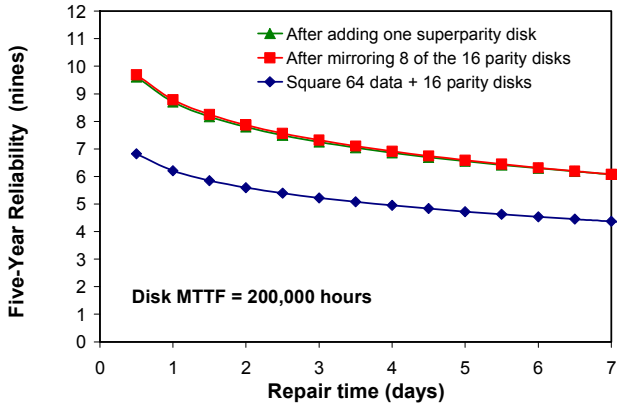

Fig. 9. Five-year reliabilities of the three array organizations for a disk MTTF of 200,000 hours.

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

$$MTTDL = \sum_{i=0}^{5} p_i^*(0) ,$$

we solve the system of Laplace transforms for $s = 0$ and a fixed value of N using the Maxima symbolic algebra package [10]. We then use this result to compute the MTTDL of our system and convert this MTTDL into a five-year reliability using the formula

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

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 focused our analysis on two-dimensional arrays with 64 data disks. Under normal circumstances, these arrays comprise 8 row parity disks and 8 column parity disks. Their space overhead is thus 16/80 = 20 percent, which is fairly reasonable. When one of these arrays experiences much higher than expected disk failure rates, we have two options to increase its fault-tolerance:

1. We can add a superparity disk, thus increasing the number of parity disks from 16 to 17. While this option is very space-efficient, it also severely limits the update bandwidth of the array.
2. We can mirror 8 of the 16 parity disks, thus increasing the number of parity disks from 16 to 24. While this second option is less space-efficient than the first one, it has much less impact on the update bandwidth of the array.

Figures 9 and 10 summarize our findings. They show the five-year reliabilities of the three configurations we investigated for average repair times varying between half a day and one week. The lower curve displays the reliability of the original array configuration with 64 data disks and

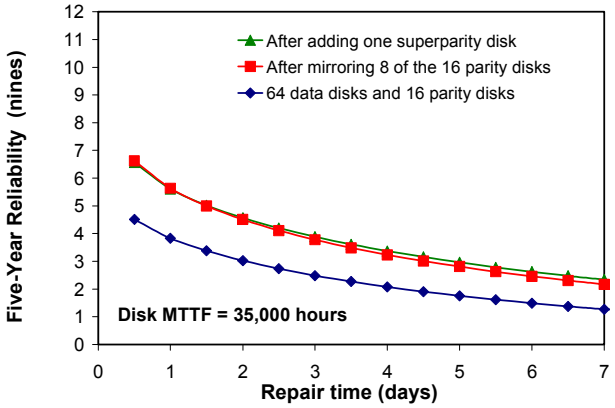

Fig. 10. Five-year reliabilities of the three array organizations for a disk MTTF of 35,000 hours.

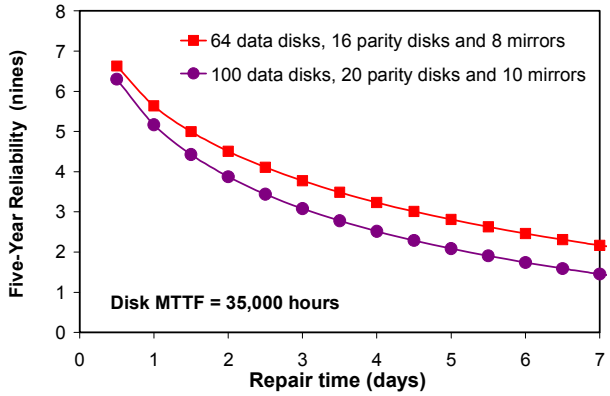

Fig. 11. Five-year reliabilities of arrays of different sizes after one half of their parity disks have been mirrored.

and 16 parity disks while the two overlapping curves on the top respectively represent the reliabilities of:

1. The original array after we added a superparity disk and
2. The original array after we mirrored 8 of its 16 parity disks.

For space considerations, we did not include the derivations of the reliabilities of the original array configuration with 64 data disks and 16 parity disks or that of the same array after we added a superparity disk. Their derivations are analogous to that of the array after we mirror 8 of its 16 parity disks and have been published elsewhere [14].

Fig. 9 displays the five-year reliabilities of the three array configurations for a disk MTTF of 200,000 hours, which corresponds to a yearly disk failure rate of 4.28 percent. These reliabilities represent what can be expected from an array built with good disks. As we can see, the original array configuration with 64 data disks and 16 parity disks performs fairly well providing five-year survival rates exceeding five nines (99.999 percent), as long as faulty disks are replaced within three days. We can thus conclude that the protection offered by the original configuration will suffice for most applications.

The situation is quite different when the disks are much less reliable. Fig. 10 displays the same values as Fig. 9, but we assume this time a disk MTTF of 35,000 hours, which corresponds to a yearly disk failure rate of 25 percent. While this failure rate is fairly high, it is neither exceptional nor confined to disks of dubious origin. As we mentioned earlier, Beach [1] reported just such a rate for a batch of 539 disks coming from a reputable manufacturer. We note that the original array configuration with 64 data disks and 16 parity disks could not achieve five nines reliability over a five year period. In addition, that reliability falls under two nines (99 percent) when disks are not replaced within four days. Conversely, either adding a superparity disk to the array or mirroring 8 of its 16 parity disks will guarantee a five-year reliability of five nines percent when the disks are replaced within 36 hours. In addition, replacing disks within four and half days still provides a five-year reliability of three nines (99.9 percent).

We also wanted to know whether the same technique would work for slightly larger arrays and considered the case of an array with 100 data disks and 20 parity disks, 10 of which have been mirrored. As Fig. 11 shows, the larger array is slightly less reliable than the smaller array as we now need to replace the disks by next day to guarantee a five-year reliability of five nines.

V. Possible Extensions

Eliminating all fatal triple failures was so far a relatively straightforward task because all these failures involved both a row parity disk and a column parity disk. As a result, mirroring either set of parity disks sufficed to eliminate all fatal triple failures. Let us now consider hardening techniques for disk array organizations that have more fatal triple failures.

Consider for instance complete two-dimensional RAID organizations [12] such that:

1. Each parity stripe contains a single parity disk;
2. All parity stripes intersect with each other;
3. Each intersection contains exactly one data disk.

Since all its parity stripes intersect, a complete two-dimensional RAID array with n parity disks will have a total of $n(n-1)/2$ data disks, that is, significantly more than the square RAID arrays we have considered so far. For instance, the complete two-dimensional RAID array of Fig. 12 has four parity disks and six data disks while a square array with the same number of parity disks would only have four data disks.

Unfortunately, this more compact organization has many more fatal triple failures than a square organization. As we can see in Fig. 12, the simultaneous failure of any three data disks located at the intersection of three parity stripes will result in a data loss. The best way to eliminate

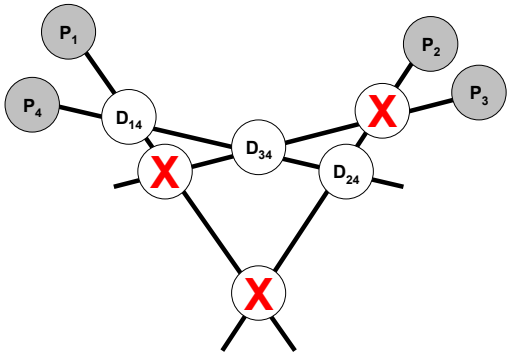

Fig. 12. A complete two-dimensional RAID array showing a new type of fatal triple failure.

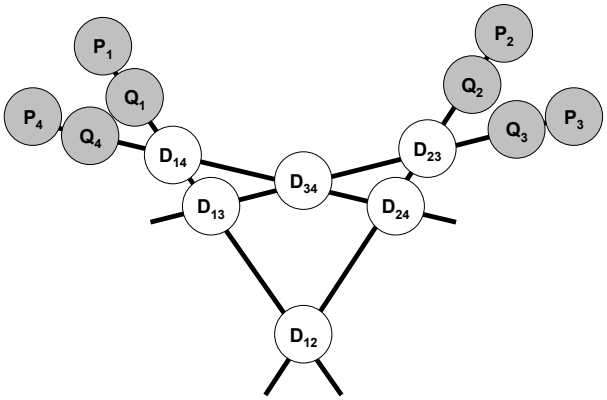

Fig. 13. A complete two-dimensional RAID array where each parity stripe is a RAID level 6 array.

these failures is to double the number of parity disks and transform each parity stripe into a RAID 6 level stripe.

Fig. 13 displays the outcome of this process. As the new array can tolerate two failures per parity stripe, the new array can recover from the simultaneous failure of any three data disks located at the intersection of three parity stripes. Since each data disk has now four parity disks, failures of any data disk and two of its parity disks cease to cause a data loss. As a result, the new array tolerates all triple disk failures. In addition, there are no fatal failure patterns [22] involving less than five disks. The minimal fatal failure pattern we observed was the failure of a data disk and its four parity disks, say, data disk D_{34} and parity disks P_3, Q_3, P_4 and Q_4. Hence, the outcome of our upgrade is an array that can tolerate all quadruple failures and some, but not all, quintuple failures. The upgraded array will thus be *significantly more reliable* than a two-dimensional RAID array of the same size with partial parity mirroring.

The impact of the upgrade on the storage overhead of the array would still be moderate as we would be starting with an array that has a smaller storage overhead than square RAID arrays. Recall that a square array with 64 data disks required 3×8 = 24 parity disks to be protected against

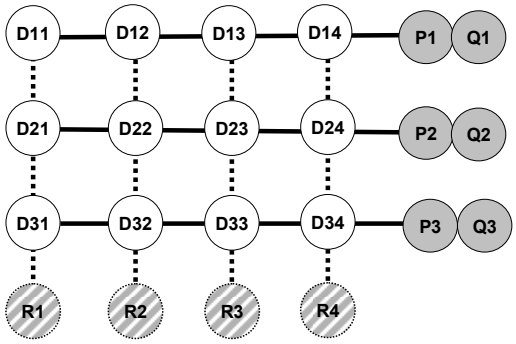

Fig. 14. A set of three RAID level 6 stripes hardened by the addition of column parity disks, R_1 to R_4. In reality the P and Q parity blocks would be distributed among the disks forming each stripe.

all triple disk failures. In contrast, a complete RAID array with 66 data disks would only require 12 parity disks. Even after doubling the number of parity disks, we would have a space overhead of 24/88 = 27 percent and the same number of parity disks as than the hardened version of the square RAID array. The major drawback of the upgrade would thus be a more complex—and potentially slower—update procedure as four parity disks would have to be updated instead of two.

Another candidate for hardening would be sets of RAID level 6 stripes. We could group all stripes of each set into a two-dimensional RAID organization. Assuming that each individual RAID level 6 stripe forms one of the row parity stripes of our new organization, each column parity stripe will include one data disk from each RAID level 6 stripe and have its own parity disk. Fig. 14 displays one such organization comprising three small RAID level 6 arrays with six disks each instead of the usual ten or twelve. It has been hardened by the addition of column parity disks, R_1 to R_4, that protect the data against all triple failures. The additional space overhead would depend on the number of RAID level 6 arrays in each array set. Larger sets would result in a smaller additional overhead but would also limit the update rate of the array.

There are strong similarities and differences between this organization and an organization merely mirroring the one set of parity disks. Both organizations arrange their data disks in a rectangular array where each data disk is placed at the intersection of one row and one column parity stripe. Both organizations require one set parity stripes to contain two parity disks per stripe while all other party stripes contain a single parity disk. Their main difference is the way the two parity disks in the row stripes are used: starting with and keeping a RAID level 6 organization for the row parity stripes results in more complex update procedures but also eliminates all fatal quadruple failures of type B and C (but not those of type A). The hardened array will thus be *somewhat more reliable* than a two-dimensional RAID array of the same size with partial parity mirroring.

At the present stage, one may wonder why our solution for two-dimensional square arrays was to mirror one of the two sets of parity disks instead of transforming each stripe of that set into a RAID level 6 stripe. Given that this approach would have eliminated most but not all fatal quadruple failures, we only observed modest improvements in the five-year reliability of the array and did not believe that these improvements would have justified the accompanying increase in complexity of the update.

VI. Conclusion

Disk failure rates vary so widely among different makes and models that it is very difficult to predict the actual failure rates of the disks we plan to use in a new disk array. We propose to handle this uncertainty by designing our storage solutions for the most probable case while incorporating in our designs the option of quickly increasing the redundancy level of the array if its components prove themselves to be less reliable than expected.

We have shown how our approach would apply to two-dimensional disk arrays with n^2 data disks and $2n$ parity disks. Should these disks prove themselves to be less reliable than expected, we would add n additional parity elements to the array and have them mirror the contents of either the n row or the n column parity disks of the array (but not both). As a result, the new array becomes able to tolerate all triple disk failures. Our solution offers the three advantages of being easy to deploy, not affecting the complexity of parity calculations, and providing a five-year reliability of 99.999 percent (five nines) for disk failure rates as high as 25 percent per year.

More work is still needed to investigate in depth the application of our approach to other disk array organizations.

Acknowledgments

A. A. and D. D. E. L. were supported in part by the National Science Foundation under awards CCF-1219163 and CCF-1217648, by the Department of Energy under award DE-FC02-10ER26017/DE-SC0005417 and by the industrial members of the Storage Systems Research Center.

References

[1] B. Beach, "What hard drive should I buy?" http://blog.backblaze.com/2014/01/21/, January 21, 2014, retrieved April 24, 2014.

[2] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EvenOdd: An efficient scheme for tolerating double disk failures in RAID architectures," IEEE Trans. on Computers 44(2):192–202, 1995.

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

[4] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," Proc. USENIX Conference on File and Storage Technologies (FAST 2004), pp. 1–14, Mar.-Apr. 2004.

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

[6] W. Gang, L. Xiaoguang, L. Sheng, X. Guangjun, and L. Jing, "Generalizing RDP codes using the combinatorial method," Proc. 7th IEEE International Symposium on Network Computing and Applications, pp. 93–100, July 2008.

[7] 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

[8] C. Huang and L. Xu, "STAR: an efficient coding scheme for correcting triple storage node failures," Proc. 4th USENIX Conference on File and Storage Technologies (FAST 2005), pp. 197–210, Dec. 2005.

[9] 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.

[10] Maxima, a Computer Algebra System, http://maxima.sourceforge.net/, retrieved August 1, 2014.

[11] J.-F. Pâris, T. Schwarz, S .J. and D. D. E. Long, "Self-adaptive archival storage systems," Proc. 26th International Performance of Computers and Communication Conference (IPCCC 2007), pp. 246–253, Apr. 2007.

[12] J.-F. Pâris, A. Amer, and T. Schwarz, S.J., "Low-Redundancy Two-Dimensional RAID Arrays," Proc. 2012 International Conference on Computing, Networking and Communications (ICNC 2012), Data Storage Technology and Applications Symposium, pp. 507–511, Jan.-Feb. 2012.

[13] J.-F. Pâris, T. Schwarz, S. J. and D. D. E. Long, "Self-adaptive archival storage systems.," Proc. 26th International Performance of Computers and Communication Conference (IPCCC 2007), pp. 246–253, Apr. 2007.

[14] 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 International Performance of Computers and Communication Conference (IPCCC 2012), pp. 324–331, Dec. 2012

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

[16] 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 2007), pp. 17–28, Feb. 2007.

[17] J. S. Plank, "A new minimum density RAID-6 code with a word size of eight," Proc. 7th IEEE International Symposium on Network Computing and Applications (NCA 2009), pp. 85–92, July 2009.

[18] J. S. Plank, "The RAID-6 liberation codes," Proc. 6th USENIX Conference on File and Storage Technologies (FAST 2008), pp. 1–14, Feb. 2008.

[19] 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 2007), pp. 1–16, Feb. 2007.

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

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

[22] T. Schwarz, S.J., D. D. E. Long and J.-F. Pâris, "Reliability of Disk Arrays with Double Parity," Proc. 19th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2013), pp. 108–117, Dec. 2013.

[23] 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 International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2009), pp. 246–256, Sep. 2009.