Title

Three-Dimensional Redundancy Codes for Archival Storage

ISBN

9780769551029

Authors

Pâris, Jehan-François
Long, Darrell DE
LAMSADE, Witold Litwin

Publication Date

2013-08-01

DOI

10.1109/mascots.2013.45

Three-Dimensional Redundancy Codes for Archival Storage

Conference Paper · August 2013

DOI: 10.1109/MASCOTS.2013.45

CITATIONS
10

READS
79

3 authors, including:

Jehan-Francois Paris
University of Houston
165 PUBLICATIONS **2,570** CITATIONS

SEE PROFILE

Darrell D. E. Long
University of California, Santa Cruz
316 PUBLICATIONS **9,286** CITATIONS

SEE PROFILE

Three-Dimensional Redundancy Codes for Archival Storage

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

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

Witold Litwin
LAMSADE
Université Paris-Dauphine
Paris, France
witold.litwin@dauphine.fr

***Abstract*—Fault-tolerant disk arrays rely on replication or erasure-coding to reconstruct lost data after a disk failure. As disk capacity increases, so does the risk of encountering irrecoverable read errors that would prevent the full recovery of the lost data. We propose a three-dimensional erasure-coding technique that reduces that risk by guaranteeing full recovery in the presence of all triple and nearly all quadruple disk failures. Our solution performs better than existing solutions, such as sets of disk arrays using Reed-Solomon codes against triple failures in each individual array. Given its very high reliability, it is especially suited to the needs of very large data sets that must be preserved over long periods of time.**

***Keywords*-RAID arrays**

I. INTRODUCTION

One of the major challenges facing the storage community is finding efficient ways to store enormous amounts of data and preserve its integrity over time periods that can span decades. There are already several archives that exceed ten petabytes. Such a ten-petabyte archive would require ten thousand one-terabyte disks. Suppose that we group these ten thousand disks into one thousand disks arrays each with ten data disks and we add enough redundant storage to each individual array to give it a 99.99 percent reliability over the lifetime of the archive. As the reliability of the whole archive is the product of the reliabilities of its constituting elements, that reliability would only be 90.5 percent. As a result, traditional solutions that worked well for much smaller data sets become inadequate.

Cost is another factor to consider. Mirroring is a popular solution for storing small to medium-size data sets. It is less effective for very large archives as it would double the hardware cost and the power consumption of an already costly storage system. At the same time, solutions that provide higher reliability with a lower space overhead become much more attractive.

We present a storage architecture that satisfies these two criteria. Conventional linear RAID arrays consist of several data disks and one parity disk. We extend that approach to a three-dimensional space and organize our data disks in such a way that each data disk participates in three distinct parity groups. As a result, the contents of these disks are protected against all triple and nearly all quadruple failures. Another advantage of our approach is its low space overhead. A three-dimensional array with n_p parity disks can protect the contents of up to $\binom{n_p}{3}$ data disks.

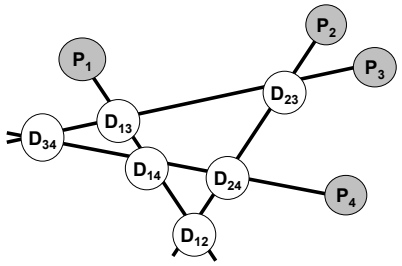

Figure 1. A two-dimensional RAID organization using four parity disks to protect the contents of six data disks against all double disk failures.

II. PREVIOUS WORK

Two-dimensional RAID arrays, or 2D-Parity arrays, were investigated by Schwarz [1] and Hellerstein et al. [2] who noted that these arrays tolerate all double disk failures but did not investigate how they reacted to triple or quadruple disk failures. 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 [3]. Pâris et al. [4] presented two-dimensional RAID arrays that reorganized themselves after a disk failure and noted that all two-dimensional RAID arrays tolerated most triple failures. More recently, Pâris et al. [5] investigated two-dimensional RAID arrays consisting of n_p parity disks and $\binom{n_p}{2}$ data disks and showed that this architecture protected the contents of the data disks against all double and most triple failures. Uehara presented several MeshRAID organizations [6, 7].

III. OUR TECHNIQUE

Before introducing three-dimensional RAID arrays, let us briefly describe the two- dimensional RAID organizations on which they are based. Consider the two-dimensional RAID organization described in Fig. 1. It consists of four parity disks, labeled P_1 to P_4, and six data disks, labeled D_{12} to D_{34}. These disks are organized in such a way that [5]:

1. Each parity disk is on a separate parity stripe;
2. All parity stripes intersect with each other;
3. These intersections contain a single data disk;
4. All data disks belong to exactly two parity stripes.

† Supported in part by Grant CCF-1219163, by the Department of Energy under Award Number DE-FC02-10ER26017/DE-SC0005417 and by the industrial members of the Storage Systems Research Center.

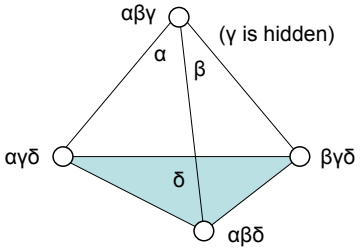

Figure 2. A three-dimensional RAID organization using four parity disks to protect the contents of four data disks against all triple disk failures. Each data disk is identified by the three parity planes to which it belongs.

Adding a fifth parity disk to the array, would allow us to form an additional parity stripe and place four new data disks on this stripe, all located at the intersection of the new parity stripe with one of the extant four parity stripes. More generally, adding an additional parity disk to an array with n_p parity disks would allow us to add one additional data disk to each of the extant n_p parity stripes disks for a total $\binom{n_p+1}{2}$ data disks and $n_p + 1$ parity disks. As all data disks belong to two distinct parity stripes, the array can recover from all double failures without losing any data. [5].

Moving from two to three dimensions is the most natural way to increase the protection afforded by two-dimensional arrays. To keep the duality between parity stripes representing segments and data disks represented by segment intersections, we introduce the notion of *parity plane*. A parity plane consists of a certain number of data disks and one parity disk that contains the exclusive or (XOR) of the contents of the data disks.

Our new organization will be defined by the following four properties:

1. Each parity plane contains a distinct parity disk;
2. All parity planes intersect with each other;
3. The intersections of three parity planes contain a single data disk;
4. All data disks belong to exactly three parity planes.

As all data disks belong to three distinct parity planes, these three-dimensional RAID arrays will be able to recover from all three failures without losing any data.

Figure 2 represents a three-dimensional RAID array with four data disks and four parity planes As we can see, the four intersecting planes α, β, γ, and δ define a tetrahedron with six edges and four summits, and each of these summits defines a data disk, to which we attach the labels of the three intersecting parity planes. Because each data disk is on three distinct parity planes, the array will be able to recover from all triple disk failures without any data loss.

Observing that all parity planes intersect with each other and each intersection of three parity planes consists of a single data disk, we can see an array with n_p parity disks—and the same number of parity planes—will hold $\binom{n_p}{3}$ data disks. Since each data disk is on three separate parity planes, each parity plane will contain $\frac{3}{n_p}\binom{n_p}{3} = \binom{n_p-1}{2}$ data disks. For instance, a three dimensional array with 6 parity disks will contain 20 data disks and each parity plane will contain 10 data disks and one parity disk.

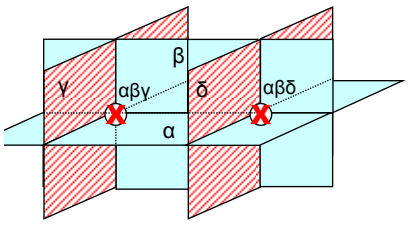

Figure 3. A pair of data disks that have failed and lost the two parity planes γ and δ they do not share. The data disks cannot recover from their two other parity planes α and β because they share both planes.

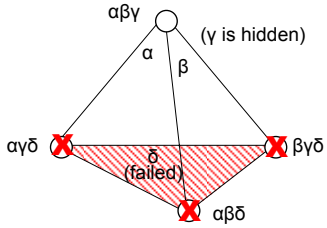

Figure 4. Three data disks that have failed and lost their shared parity plane δ. They cannot recover from their three other parity planes α, β and γ. Each data disk is identified by the three parity planes to which it belongs.

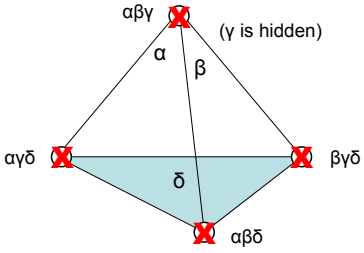

Figure 5. A fatal quadruple failure where all four failed disk occupy the four summits of a tetrahedron formed by four parity planes.

Let us now consider how three-dimensional arrays react to quadruple failures. As long as these failures involve unrelated disks, the array will be able to recover without any data loss. The sole exceptions are:

1. The failure of a data disk and its three parity disks: both the data on the data disk and the parity information needed to reconstruct them is lost.
2. The failure of two data disks and the two parity disks they do not share: since the data disks share their two other parity disks, we cannot use them to recover the contents of the two data disks. Figure 3 illustrates this case.
3. The failures of three data disks occupying the three summits of one side of a tetrahedron formed by four parity planes plus the parity disk of the parity plane that contains the three data disks: recovery is prevented because the data disks also share their other parity planes. Figure 4 illustrates this case.
4. The failure of four data disks such that none of the four failed disks belongs to a parity plane that is not shared with one of the three other disks: recovery is then impossible because each parity plane contains a single parity disk and can only recover from a single disk failure within the parity plane. We have to distinguish here among three possible scenarios:

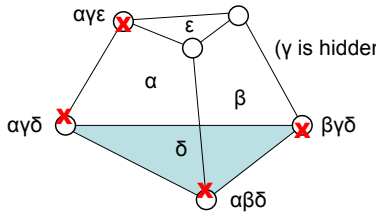

Figure 6. A quadruple failure where the four failed disks occupy four summits of a truncated triangular pyramid formed by five parity planes. Node αγε will be able to recover as it is the sole failed disk that is in parity plane ε.

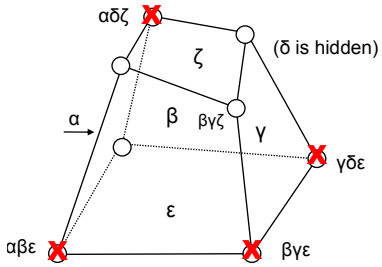

Figure 7. A quadruple failure where all four failed disks occupy four summits of a hexahedron formed by six parity planes. Node αδζ will be able to recover, as it is the sole node that belongs to the parity plane ζ.

a. The four failed data disks are the summits of a tetrahedron formed by four parity planes: the four data disks will share four parity planes and each of these four planes will contain three failed disks. Figure 5 illustrates this case.

b. The four failed data disks are at one of the six summits of a truncated triangular pyramid formed by five parity planes and none of the four failed disks belongs a parity plane that is not shared with one of the three other disks. This is the case whenever three of the four failed disks and the fourth failed disk on the opposite side will be able to recover because that failed disk belongs to a parity plane it does not share with any of the three other disks. As shown in Fig. 6, configurations that place three of the failed disks at either the top or the bottom of the truncated pyramid, and the fourth failed disk on the opposite side will be able to recover because that failed disk belongs to a parity plane it does not share with any of the three other disks.

c. The four failed disks are at four of the eight summits of a hexahedron formed by six parity planes and none of the four failed disks belongs to a parity plane that is not shared. As shown on Fig. 7, the array can recover if one of the four failed disks belongs to a parity plane that it does not share with any of the three other failed disks.

Let $n_d = \binom{n_p}{3}$ denote the number of data disks in a three-dimensional array with n_p parity disks. Out of the $\binom{n_p+n_d}{4}$ possible quadruple failures the array can experience, we can enumerate:

TABLE I. SUMMARY OF THE PROPERTIES OF SMALL TO MEDIUM-SIZE THREE-DIMENSIONAL RAID ARRAYS.

Parity disks	*Data disks*	*Total*	*Space Overhead*	*Disks per plane*	*Tolerated faults*	*Coding overhead*
3	1	4	0.750	1	3.000	1.333
4	4	8	0.500	3	3.929	1.018
5	10	15	0.333	6	3.982	1.256
6	20	26	0.231	10	3.993	1.503
7	35	42	0.167	15	3.997	1.751
8	56	64	0.125	21	3.998	2.001
9	84	93	0.097	28	3.999	2.251

1. n_d distinct failures of a data disk and its three parity disks;
2. $\binom{n_p}{2}$ distinct failures of two data disks sharing two parity planes plus their two other parity disks;
3. $4\binom{n_p}{4}$ distinct failures of three data disks forming the summits of one of the four sides of a tetrahedron plus the parity disk of the parity plane that contains the three disks;
4. $\binom{n_p}{4}$ distinct failures of four data disks forming the summits of a tetrahedron;
5. $9\binom{n_p}{6}$ distinct fatal failures of four data disks placed at the summits of a truncated pyramid: there are $\binom{n_p}{5}$ distinct truncated triangular pyramids formed by the intersection of five parity planes and 6 of the $\binom{6}{4}$ possible positions for the four failed disks are safe.
6. $14\binom{n_p}{6}$ distinct fatal failures of four data disks placed at the summits of a hexahedron: there are $\binom{n_p}{6}$ distinct truncated triangular pyramids formed by the intersection of five parity planes and 56 of the $\binom{8}{4}$ possible positions for the four failed disks on a given pyramid are safe.

As a result, the fraction of quadruple failures that will result in a data loss is

$$\alpha = \frac{n_d + \binom{n_p}{2} + 5\binom{n_p}{4} + 9\binom{n_p}{5} + 14\binom{n_p}{6}}{\binom{n_p+n_d}{4}}.$$

This ratio quickly decreases with the size of the array: it is already less than 4 percent for an array with 15 disks and becomes less than 0.5 percent for all arrays with 43 disks or more. As a result all three-dimensional arrays with 15 disks or more tolerate nearly all quadruple failures.

Table I summarizes the properties of small to medium-size three-dimensional arrays. We define the space overhead as the fraction of space occupied by the parity data and the coding

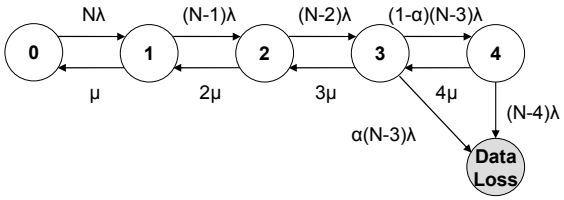

Figure 8. State transition probability diagram for a three-dimensional RAID array with N disks.

overhead as by dividing the number of parity disks by the average number of disk failures the array can tolerate.

As we can see, three-dimensional RAID arrays with 26 to 64 disks offer an attractive combination of relatively low storage and coding overheads.

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 mean time to data loss (MTTDL) of the storage system, which is the approach we will take here.

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, the repair process will be performed in parallel on those disks. We assume that disk failures are independent events and times between consecutive failures for a given device 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.

Figure 8 displays the state transition probability diagram for a three-dimensional RAID array with n_p parity disks and n_d data disks for a total of $N = n_p + n_d$ disks. State <0> is the original state where all N disks are operational. 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>, and a third failure would bring the system to state <3>. As we have seen, only a fraction α of all possible quadruple failures will result in a data loss. Hence the two failure transitions from state <3> are:

1. A transition to the failure state with rate $\alpha(N-3)\lambda$
2. A transition to state <4> with rate $(1-\alpha)(N-3)\lambda$.

As long as the total number of nodes in the array remains small, the possibility that the array will experience a quintuple failure during its lifetime will remain very small. We can thus safely neglect the probability that the array will survive a quintuple failure and assume that all quintuple failures will result in a data loss transition from state <4> to the failure state.

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 state <0>.

As we did in our previous studies [5], we compute first the Laplace transforms of the system of differential equations describing our model, solve this system and use the formula

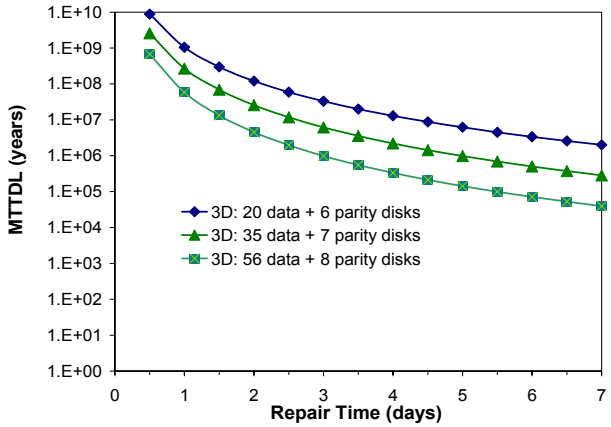

Figure 9. Mean Times To Data Loss of three-dimensional arrays with respectively 26, 42 and 64 disks.

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

to compute the MTTDL of our system.

For instance, the MTTDL of a three-dimensional RAID array with 6 parity disks and 20 data disks is

$$\frac{133735225\lambda^4 + 11846961\mu\lambda^3 + 1001589\mu^2\lambda^2 + 61603\mu^3\lambda + 1950\mu^4}{7800\lambda^4(82225\lambda + 178\mu)}$$

Figure 9 displays on a logarithmic scale the MTTDLs achieved by three-dimensional arrays with respectively 26, 42 and 64 disks. We assumed a disk failure rate λ of one failure every one hundred thousand hours, which is slightly less than one failure every eleven years. This rate is at the high end of the failure rates observed by Pinheiro et al. [8] as well as Schroeder and Gibson [9]. MTTDLs are expressed in years and repair times in days.

As we can see, the three arrays exhibit MTTDLs that exceed ten million years when failed disks are promptly replaced and one million years when the replacement process takes several days. While these very large MTTDLs may appear irrelevant, we can use them to compute the probability that the array will lose no data during its lifetime. Assuming an array lifetime of five years, and observing that array long-term failure rates do not significantly differ from their average failure rates over their first five years, we can convert the MTTDLs into reliabilities using the formula

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

where d is a five-year interval expressed in the same units as the MTTDL. We can then see that a MTTDL of ten million years corresponds to a 0.9999995 probability of no data loss over a five year interval while a MTTDL of one million years correspond to a 0.999995 probability of no data loss over the same time interval.

We also compared these MTTDLs with those that could be obtained by two other disk array organizations. The two benchmarks we selected were:

1. A pair of disk arrays with 10 data disks and 3 parity disks each: this organization offers the same storage capacity and the same storage overhead as a three-dimensional RAID array with 20 data disks and 6 parity disks.

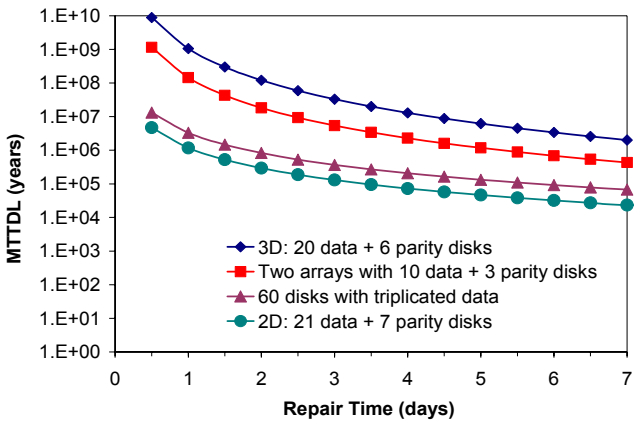

Figure 10. Compared Mean Times To Data Loss of (a) a three-dimensional RAID array with 20 data disks and six parity disks, (b) a pair of RAID arrays with 10 data disks and 3 parity disks each, (c) 60 disks containing triplicate data and (d) a two-dimensional RAID array with 21 data disks and 7 parity disks. The first three configurations have the same storage capacity.

2. A single disk array with 60 disks using three-way mirroring to store the equivalent content of 20 data disks.
3. A two-dimensional RAID array with 21 data disks and 7 parity disks [5].

Figure 10 summarizes our results. As we can see, the three-dimensional RAID organization achieves MTTDLs that are 4.5 to 7.5 times better than those obtained by the pair of RAID arrays with triple parity. Both organizations provide MTTDLs much higher than those obtained by the three-way mirroring organization and the two-dimensional RAID array. This should not surprise us as three-way mirroring and two-dimensional RAID arrays only protect data against double disk failures while the two other organizations can tolerate triple disk failures without data loss.

V. Implementation Considerations

Achieving the highest possible reliability for a given redundancy level is not the sole factor to consider when selecting a disk array organization. If this were the case, all disk arrays would use Reed-Solomon codes. There are instead other factors to consider such as update rates and the complexity of data recovery operations.

Update rates are essentially limited by the complexity of the update operations and by the existence of potential bottlenecks. All storage arrays that protect against triple disk failures require all updates to be recorded on at least four disks. Three-dimensional RAID arrays do not differ in that sense from any other solution including those using Reed-Solomon codes. The main difference is that they comprise distinct data disks and parity disks while most other solutions allow each disk to hold both data and parity information. Since all updates must be propagated to three parity disks, the update rates of three-dimensional RAID arrays cannot exceed one third of the aggregate update bandwidth of their parity disks. As a result, these arrays are best suited to applications with low update rates and high reliability requirements such as archival storage systems.

Data recovery offers a different picture. Whenever a conventional RAID array loses a data disk, it needs to read the contents of all the other data disks in the same parity stripe as the failed disk plus one of the parity disks of that stripe. The process is essentially sequential and often very slow. This is not the case for three-dimensional RAID arrays. Since each data disk belongs to three distinct parity planes, we can speed up data recovery by accessing in parallel these three parity planes and dividing recovery tasks among them. Installations with three or more spare disks on hand could achieve an even higher speedup by reconstructing the lost data on three separate disks [10].

VI. Conclusions

We have presented a three-dimensional RAID organization where each data disk belongs to three parity planes and each parity plane contains a single parity disk. As our organization requires all parity planes to intersect and prevents more than three planes from sharing a common intersection, we can place a maximum number of data disks at each intersection of the three parity planes. The outcome is an organization that requires n_p data disks to protect its contents against all triple failures and at least 98 percent of quadruple disk failures, thus performing much better than organizations storing all data in triplicate and significantly better than other triple-parity RAID organizations. Its sole limitation is its relatively low update bandwidth, which restricts its use to applications with low update rates and high reliability requirements such as archival storage systems.

More work is still needed to evaluate the impact of correlated failures on three-dimensional array reliability and investigate other three-dimensional organizations.

References

[1] T. J. E. Schwarz, “Reliability and Performance of Disk Arrays,” Ph.D. Thesis, Dept. of Computer Science and Engineering, University of California, San Diego, 1994.

[2] 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.

[3] 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

[4] J.-F. Pâris, T. J. E. Schwarz and D. D. E. Long, “Self-adaptive archival storage systems,” Proc. 26th Int. Performance of Computers and Communication Conf., pp. 246–253, Apr. 2007.

[5] J.-F. Pâris, A. Amer, and T. J. E. Schwarz, “Low-Redundancy Two-Dimensional RAID Arrays,” Proc. 2012 Int. Conf. on Computing, Networking and Communications, Data Storage Technology and Applications Symp., pp. 507–511, Jan.-Feb. 2012.

[7] M. Uehara, “Design and implementation of 3D MeshRAID in virtual large-scale disks,” Proc. 3rd Int. Conf. on Intelligent Networking and Collaborative Systems, pp. 490–495, Nov.–Dec. 2011.

[8] E. Pinheiro, W.-D. Weber and L. A. Barroso, “Failure trends in a large disk drive population,” Proc. 5th USENIX Conf. on File and Storage Technologies, pp. 17–28, Feb. 2007.

[8] M. Uehara, “Design and Implementation of Mesh RAID with multiple parities in virtual large-scale disks,” Proc. 26th IEEE Int. Conf. on Advanced Information Networking and Applications, pp. 67–72, Mar. 2012.

[9] 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 Conf. on File and Storage Technologies, pp. 1–16, Feb. 2007.

[10] I. Corderi, T. J. Schwarz, A. Amer, D. D. E. Long and J.-F. Pâris, “Self-Adjusting Two-Failure Tolerant Disk Arrays,” Proc. 5th Int. Workshop on Petascale Data Storage, Nov. 2010.