Increased Reliability with SSPiRAL Data Layouts

Ahmed Amer
University of Pittsburgh
amer@cs.pitt.edu

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

Jehan-François Pâris
University of Houston
paris@cs.uh.edu

Thomas Schwarz
Santa Clara University
tjschwarz@scu.edu

Abstract

We evaluate the reliability of storage system schemes consisting of an equal numbers of data disks and parity disks where each parity disk contains the exclusive or (XOR) of two or three of the data disks. These schemes are instances of Survivable Storage using Parity in Redundant Array Layouts (SSPiRAL). They have the same storage costs as mirrored organizations and use very simple parity schemes. Through a novel dynamic analysis of the likelihood of data losses, we show that these schemes are one hundred thousand to a million times less likely to lose data than a comparable mirrored organization. We also found that schemes where each parity disk contains the exclusive or of three data disks performed much better than schemes where each parity disk contains the exclusive or of only two data disks.

1. Introduction

The volume of digital data is growing, as is the need to build reliable storage infrastructure. In a recent study, the volume of digital data generated in 2002 was quoted at over 5 Exabytes, 92% of which was written to magnetic disk drives [16, 17]. Such growth will inevitably be reflected in the storage demands of data servers, as well as the storage demands of consumers and producers of such content. This rate of growth is only compounded by the desire – and frequently the need – to retain this data, and will inevitably result in the accelerated growth of the number of data storage devices and servers. More components implies an increased need to protect against the failure of individual components. Data storage devices have a recent history of impressive growth in capacity, this growth alone (assuming it is maintained) could easily be consumed solely by the desire to retain data, and cannot mitigate the increase in storage nodes and devices. Redundant storage schemes are an obvious solution to increasing reliability, and such applications commonly employ one of two strategies: a combination of replication and parity applied efficiently across an array of devices, or a failure-recovery scheme based on erasure coding.

Computational efficiency is important when implementing redundancy schemes for disks, and so parity is particularly appealing due to its ease of computation. There are also combinations of the two approaches, but typically parity schemes tolerate only a small number of component failures, while erasure codes tend to be expensive to implement. Excellent parity-based erasure codes and layout schemes have been devised [7, 22], but prior art has focused primarily on aiming to survive a specific number of device failures. We present a scheme, and analytic evaluation, focused on reducing the likelihood of data loss. While SSPiRAL layouts are efficient parity-based layouts that are capable of exploiting heterogeneity in the underlying devices, we set aside such functional advantages and present a novel analytical evaluation that demonstrates its ability to dramatically reduce the likelihood of data loss in the face of device failures. Our analysis avoids the pitfalls inherent in estimating MTTDLs of hundreds and thousands of years, which can be misleading for systems that are only used for a few years. In Section 2 we describe SSPiRAL layouts and in Section 3 we present our analytical results. We discuss related works and conclusions in Sections 4 and 5.

2. SSPiRAL Description

Data layouts aimed at increasing storage reliability have employed parity, erasure coding, or some combination of the two to make use of excess storage capacity in avoiding data loss. Traditionally, such layouts treat individual devices as independent and equal units, *e.g.*, in an individual RAID array all disks are considered to be equals, and efforts are made to distribute load and responsibility equally among these disks. SSPiRAL (Survivable Storage using Parity in Redundant Array Layouts) [2] is a redundant data layout scheme that deliberately considers the possibility that individual disks are not necessarily equal. This can allow the exploitation knowledge regarding the relative likelihood of failure of individual components in a storage system, but our evaluation makes the conservative assumption that there is no such knowledge available, and under such conditions, SSPiRAL layouts still offer a dramatic reduction in data loss likelihood. SSPiRAL's goal is to provide the most effective data layout across devices, with the aim of minimizing the probability of data loss, and while being defined by parameters that represent simple real-world constraints.

Every SSPiRAL layout is defined by three parameters: the degree of the system, the number of devices available, and the *x*-order where the value *x* represents the number of devices that contribute to an individual parity calculation. The degree of a SSPiRAL layout is the number of distinct data devices, representing how many disks the data will be distributed across (*e.g.*, for load balancing or to exploit parallelism). A SSPiRAL arrangement that uses a fixed value for *x* is described as a fixed-order array, and in such a layout each parity device holds data that is computed as the result of an XOR operation across exactly *x* data devices. We consider only fixed-order SSPiRAL layouts.

To build a SSPiRAL layout, we start with the degree of the layout. This is effectively the number of devices contributing to the data storage capacity of the overall system. From this point, we can impose a constraint on the maximum effort and bandwidth required for parity calculations (by setting the *x*-order of the system) or by setting a limit on the amount of redundant storage we wish to contribute to reducing the likelihood of data loss (by setting the number of parity nodes). These parameters are related, as a decision to contribute only a single parity node would necessitate the participation of all data nodes in the parity computed at that node. This is effectively a simple RAID scheme with a single parity disk. Adding more parity nodes would give us the freedom to choose between a maximum *x*-order that is equivalent to the degree of the system or a smaller value. Conversely, had we started with a fixed *x*-order, this would have imposed a restriction on the minimum number of parity nodes to build a complete SSPiRAL layout. For example, a SSPiRAL arrangement of degree 3 and *x*-order 2 would use no more than two nodes to build a parity node, and would need a set of six nodes to build a complete layout. Figure 1 shows a SSPiRAL layout of degree three and $x = 2$, alongside a mirrored layout that uses the same number of disks.

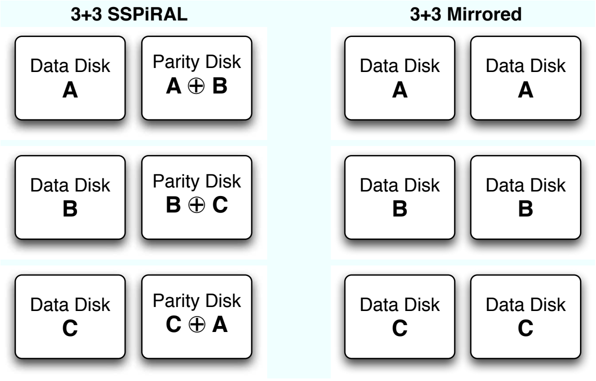

Figure 1. *3+3 SSPiRAL Layout vs. 3+3 Mirrored Disk.*

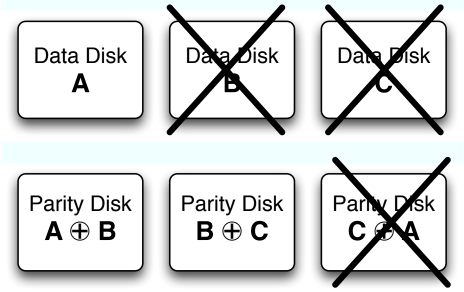

Figure 2. *SSPiRAL data layout and the loss of three nodes.*

An interesting strength of a SSPiRAL layout can be demonstrated through Figure 2, which shows the loss of three of our six devices. In spite of this loss, it is possible to recover all lost data nodes. While a mirrored array *c*an survive the loss of three nodes, it loses data if the two disks containing the same data fail. There is *no* combination of two node losses that will cause the SSPiRAL layout in Figure 2 to lose data. This increased resilience has been obtained at the expense of involving two data devices for the creation of redundant parity data on each individual parity device. Intuitively, increasing the *x*-order of a layout would seem to reduce the likelihood of data loss by increasing the paths available for reconstructing lost nodes. This intuition is supported by the impact of *x*-order we observe in Section 3.

3. Reliability Analysis

We provide an analysis of the reliability of SSPiRAL layouts, and compare them to the reliability of mirrored disk layouts. We evaluate the effects of varying the number of devices and the *x*-order of the SSPiRAL layouts. All SSPiRAL layouts considered use 50% redundancy to allow a fair comparison to the mirrored layouts. To compare the layouts we calculate the probability of data loss after a fixed number of years. This comparison assumes an expected rate of failure for the disk devices, but we do not assume any *a priori* knowledge of these rates for the SSPiRAL layouts. While a SSPiRAL layout can conceivably take advantage of differences in the expected reliability of individual devices (*e.g.*, through knowledge of a device's age or SMART data), we assume all devices to have an equal likelihood of failure.

Estimating the reliability of a given storage system means evaluating the probability $R(t)$ that the system will operate correctly over the time interval $[0,t]$ given that it operated correctly at time $t = 0$. While this approach is straightforward, it requires solving a potentially complex system of linear differential equations. As a result, many studies characterize the reliability of storage systems by their Mean Time To Data Loss (MTTDL), which only requires the solution of a system of linear equations. Unfortunately, MTTDLs of hundreds and thousands of years can be misleading for systems that are only used for a few years. Such systems will operate over their lifetime under conditions very close to their initial state where all storage devices were operational. As a result, they will experience significantly lower failure rates than those predicted using their MTTDLs. It is therefore essential to consider the dynamic behavior of each storage system over its actual lifetime.

3.1. SSPiRAL Array Layouts

Building a completely accurate state-transition diagram for a SSPiRAL array exceeds the limitations of this paper as we would have to distinguish between failures of data disks and failures of parity disks. These distinctions are necessary for complete accuracy since complexity of recalculating data previously stored in a lost drive differs. Instead, we abstract from these details and aggregate states as much as possible. We capture a system with i failed disks in a state S_i. We have a repair transition from State S_i to State S_{i-1}, $i \geq 1$, which is taken with rate $i \cdot \mu$, where μ is the inverse of the average repair time. Thus, our model assumes independent repairs of any failed devices. The repair time itself is composed of the time to detection, issue of the service call and wait for the replacement of the failed device followed by the reconstruction of the data previously stored in the failed disk. The latter component is typically several hours since for example a 1 TB disk is fully read at 10MB/sec in approximately 28 hours. It increases proportionally with the size of the disk and decreases inversely proportionally with the read/write rate. We have also failure transitions that leave State S_i with a combined rate of $(N-i)\lambda$. Partially or totally, a failure transition leads to the next State S_{i+1}, complemented by a transition to the Failure State, which is absorbing.

Our model is limited by the Markovian assumption of independent repairs and failures and by the modeling of repair in particular. Nevertheless, models of this type have been confirmed by simulation to be reasonably accurate.

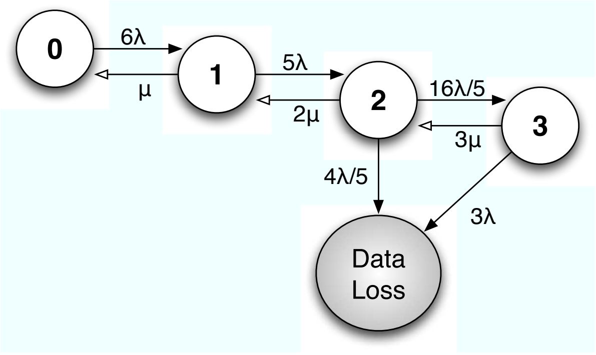

Figure 3. *Markov model for the $3+3$ SSPiRAL layout (with $x = 2$).*

3.1.1. The $3+3$ SSPiRAL Array ($x = 2$)

The $3+3$ SSPiRAL array has an x value of two (two disks contribute to each parity disk) and encompasses six disks. Its layout is given in Figure 2. Clearly, loss of four disks (or more) must lead to data loss (We are of course assuming that all data disks indeed contain data). A case by case distinction shows that there is never data loss if any two disks have failed and that in 4 out of the $\binom{6}{3} = 20$ ways in which three out of six disks can fail data loss occurs. In more detail, data loss occurs if

1. all data disks have failed

2. a data disks and the two parity devices containing its data have failed.

As a result, we have a state transition from S_3 to S_4 taken with rate $\frac{16}{20} \cdot 4 \cdot \lambda = \frac{16}{5}\lambda$ and a transition for S_3 to the absorbing (data loss) state with rate $\frac{4}{20} \cdot 4 \cdot \lambda = \frac{4}{5}\lambda$. The complete Markov model is in Figure 3.

Table 1. Data loss probability with various disk MTBF $1/\lambda$ and average repair time $1/\mu$ for the $3+3$ SSPiRAL array.

MTBF	MTTR	4	5	20	100
hours		year	year	year	year
50,000	30	3.02E-06	3.78E-06	1.51E-05	7.56E-05
10^5	30	3.78E-07	4.73E-07	1.89E-06	9.46E-06
10^6	30	3.78E-10	4.73E-10	1.89E-09	9.47E-09
50,000	100	3.34E-05	4.17E-05	1.67E-04	8.37E-04
10^5	100	4.18E-06	5.23E-06	2.10E-05	1.05E-04
10^6	100	4.19E-09	5.24E-09	2.10E-08	1.05E-07

3.1.2. The $4+4$ SSPiRAL Array ($x = 2$)

For direct comparison purposes, we consider the SSPiRAL array with

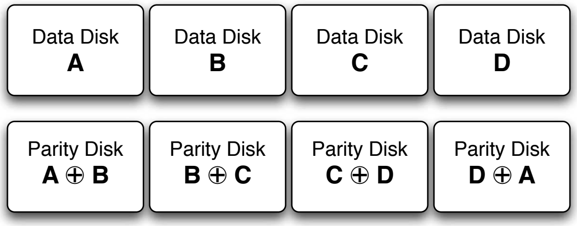

Figure 4. *The $4+4$ SSPiRAL layout (with $x=2$).*

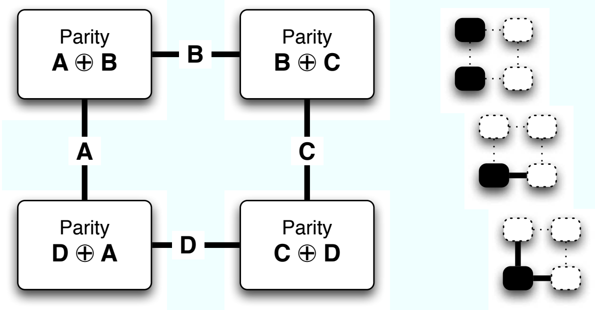

Figure 5. *Graph representation of the $4+4$ SSPiRAL layout with $x=2$ (on the left), and “neighbor”, “club,” and “two neighbor double club” patterns (on the right).*

$x=2$ and eight disks (Figure 4). To study its data survival, we use a technique similar to that developed by Hellerstein *et al.* [9] and later expanded by [11] that is based on interpreting 2-failure correcting layouts using parity calculations as a type of mathematical design called configuration (see [6]). The dual is then a regular graph. In this representation, vertices are parity disks and edges are data disks. An edge is connected to a graph if the corresponding data disk contributed to the parity. The result of this representation is in Figure 5.

Table 2. Data loss probability for the SSPiRAL array with $x=2$ and 8 disks.

MTBF	MTTR	4	5	20	100
hours		year	year	year	year
50,000	30	3.02E-06	3.78E-06	1.51E-05	7.56E-05
10^5	30	3.78E-07	4.72E-07	1.89E-06	9.46E-06
10^6	30	3.78E-10	4.73E-10	1.89E-09	9.47E-09
50,000	100	3.33E-05	4.17E-05	1.67E-04	8.36E-04
10^5	100	4.18E-06	5.23E-06	2.10E-05	1.05E-04
10^6	100	4.19E-09	5.24E-09	2.10E-08	1.05E-07

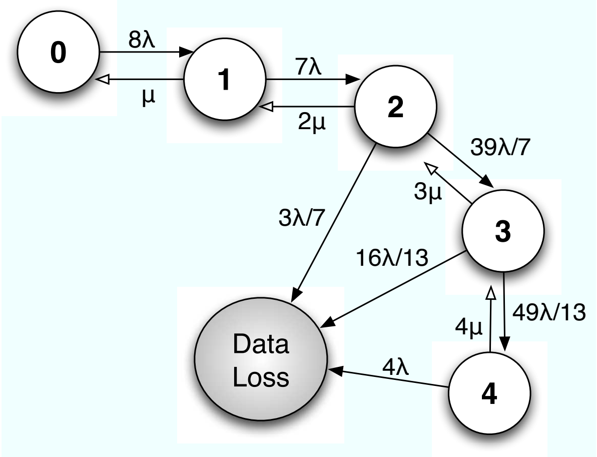

Figure 6. *Markov model for the $4+4$ SSPiRAL layout (with $x=2$).*

We now model the loss of one or more devices as a subgraph of this graph. We can reconstruct lost data by using parity calculations, which we can represent by the following graph theoretical operations. Given an edge and an adjacent vertex, we can reconstruct the other vertex. For example, given A and (A ⊕ B), we can reconstruct B. Given two adjacent edges, we can reconstruct the vertex at the intersection between these edges. For example, given A and B, we can reconstruct (A ⊕ B). Operations that are more complicated do not reconstruct additional data. As a consequence of this insight, a loss pattern leading to data loss must contain a cycle of adjacent edges (in our case the loss of A, B, C, and D) or a path consisting of edges and vertices that starts and ends in a vertex. For example, A⊕ B, A, A⊕ D is a (minimal) loss pattern with data loss. We can now calculate the probabilities that a failure leads to data loss. We use the now standard notations for the Markov model. In addition to the absorbing, data loss state, we have states S_i representing the system when i disks have failed. If there are none, one, or two failures, no data loss occurs. The three loss patterns with data loss consist of two edges with the connecting edge. Hence, the chances are $\frac{4}{56}=\frac{1}{14}$. Since the total rate of failure transitions out of S_2 is 6λ, the system transitions from S_2 at a rate $\frac{3}{7}\lambda$ to the data loss state and at rate $\frac{39}{7}\lambda$ to S_3.

We now calculate the data loss probability for the fourth failure. Assume now that we are in one of the 52 cases that does not represent data loss after failure of three devices. The failure of an additional device only results in data loss if it creates either a pattern vertex-edge-vertex or a cycle edge-edge-edge-edge. 4 out of the 52 cases consist of three edges. In this case, failure of the data disk representing the other edge leads to data loss. This happens with probability $\frac{1}{5}$. Some of the 52 cases representing failure of three devices contain a single club pattern formed by a vertex, an adjoining edge, but not the other adjoining edge. We can pick the vertex in four different ways and then one of the edges. The other failed device must not be the other edge, hence we can pick this in four different ways. This gives us 32 possibilities. The double club consists of a vertex and two adjoining edges, for this, we have four possibilities. We also have the “two neighbor” pattern, consisting of two adjacent vertices with the other failure represented not by the combining edge. There are eight cases that contain both a club and a two neighbor pattern. There are eight cases of a two neighbor pattern that does not contain a club nor the three failure pattern. Of these, four are made up of three vertices.

We now calculate the probability that an additional failure leads to data loss. In the 24 cases of a single club without a two neighbors pattern, only failure of the data disk represented by the other vertex of the clubs edge leads to data loss. In the four cases of the three vertices, three out of the five possibilities for an additional failure leads to data loss. In the four cases of the two neighbors without club and three vertices (*e.g.*, pattern A ⊕ D, C, A ⊕ B, the "face," possibly rotated), only one out of the five possibilities for further failure leads to data loss. In the eight cases of a club with two neighbors, two out of the five possibilities for further device failure lead to data loss. In the four cases of the double club, two out of the five possibilities lead to further failure.

In total, the fourth failure induces data loss with probability $\frac{24\cdot1+4\cdot3+4\cdot1+8\cdot2+4\cdot2}{52\cdot5}=\frac{16}{65}$. Since the total rate of failure transitions out of S_3 is 5λ, the transition rate from S_3 to the absorbing state is $\frac{5\lambda\cdot64}{52\cdot5}=\frac{16}{13}\lambda$ and from S_3 to S_4 $\frac{49}{13}\lambda$. We give the Markov model in Figure 6.

3.1.3. The $4+4$ SSPiRAL Array ($x=3$) We present the layout of the $4+4$ SSPiRAL array with $x=3$ in Figure 7. The modeling of the $4+4$ array is quite similar to the previous one. A case-by-case enumeration shows that there is no data loss if up to three disks have failed. An information theoretical argument shows that loss of five disks needs to lead to data loss. We consider the remaining case (failure of four disks) in more detail. We make a case distinction according to the number of data disks.

1. One lost data disk: Assume that data disk A (see Figure 7) has failed. Three parity drives have also failed and one remains available. If this one is (A⊕B⊕C), (C⊕D⊕A), or (D⊕A⊕B), then we can reconstruct the data previously in A. In the remaining case, all disks with contents reflecting A are lost and data loss is inevitable. Hence, we have data loss in 4 of the 16 cases where one data disk is lost.

2. Two lost data disks: First, we assume that two neighboring data disks in Figure 7. Let these be A and B. Two of the parity drives are also available. If (B⊕C⊕D) or (C⊕D⊕A) are among them, then we achieve directly the contents of B and A, respectively. In the remaining case, C, D, (A⊕B⊕C), and (D⊕A⊕B) are available. Since any reconstruction has to use XORing as a primitive operation and since C, D, (A⊕B⊕C), (D⊕A⊕B), (C⊕D) are the elements of a set closed under XORing, the contents of A and B remain unavailable. Of the 4 subcases, four lead to data loss. Second, we assume that two non-neighbors in Figure 7 are available. Let these be A and C. If B⊕C⊕D or D⊕A⊕B are available, we obtain with B and D directly C or A respectively and hence A and C from the other parity drive contents. This leaves the case where B, D, A⊕B⊕C, and D⊕A⊕B are available. Taking XORs of these four available objects, we obtain a set that additionally contains B⊕D and A⊕C, but that is closed under further taking of pair-wise parity. Hence, of the 2×6 sub-cases, two lead to data loss.

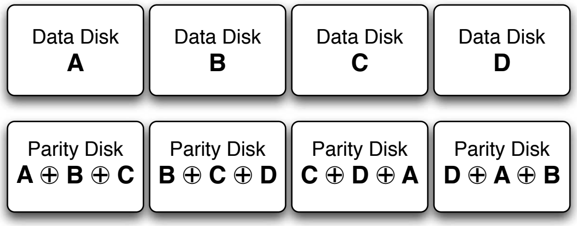

Figure 7. *The* $4+4$ *SSPiRAL layout with eight disks and* $x=3$.

3. Three lost data disks: Assume that disk A is available. We can obtain the remaining data disks contents in three of the four cases. For example, if we have A, (A⊕B⊕C), (B⊕C⊕D), and C⊕D⊕A available. Then also B = (B⊕C⊕D) ⊕ (A⊕B⊕C) ⊕ A, C = (A⊕B⊕C) ⊕ (B⊕C⊕D) ⊕ (C⊕D⊕A), D = (B⊕C⊕D) ⊕ (A⊕B⊕C) ⊕ A. But if A, (A⊕B⊕C), (C⊕D⊕A), and (D⊕A⊕B) are available, then by taking all possible pair-wise parity, we obtain the set S = A, A⊕B⊕C, C⊕D⊕A, D⊕A⊕B, B⊕C, C⊕D, D⊕B which is closed under this operation. Hence, in this case the array suffers data loss. *In toto*, of the 16 subcases, 4 lead to data loss.

4. Four lost data disks: Since A = (A⊕B⊕C) ⊕ (C⊕D⊕A) ⊕ (D⊕A⊕B), B = (D⊕A⊕B) ⊕ (A⊕B⊕C) ⊕ (B⊕C⊕D), C = (A⊕B⊕C) ⊕ (B⊕C⊕D) ⊕ (C⊕D⊕A), D = (B⊕C⊕D) ⊕ (C⊕D⊕A) ⊕ (D⊕A⊕B), there is no data loss.

To summarize, out of a total of $\binom{8}{4}=70$ ways for four out of eight disks to fail, 14 lead to data loss. We can now use our in-sight to calculate the Markov model given in Figure 8. There is a combined rate of 5λ of failure transitions out of State S_3. The rate of the transition from S_3 to the absorbing state is $\frac{14}{70}\cdot5\lambda=\lambda$ and of the transition from State S_3 to State S_4 is $\frac{56}{70}\cdot5\lambda=4\lambda$.

We give the survival rates after 4 and 5 years in Table 3. Compared with the SSPiRAL array also with eight disks

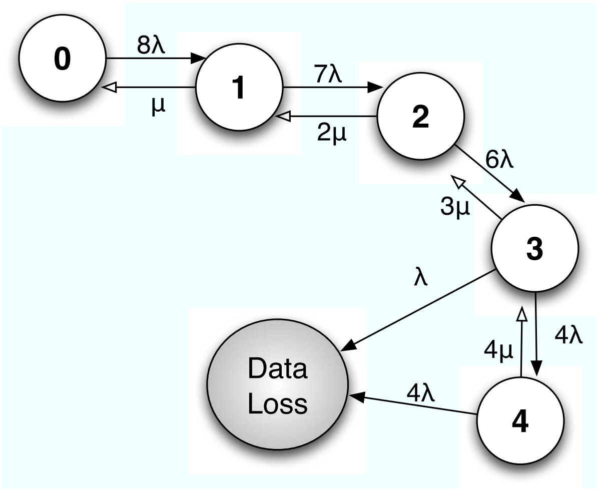

Figure 8. *Markov model for the* $4+4$ *SSPiRAL layout (with* $x = 3$*).*

but with $x = 2$, the numbers are better by about three powers of ten.

Table 3. Data loss probability with various disk MTBF $1/\lambda$ **and average repair time** $1/\mu$ **for the** $4+4$ **SSPiRAL array.**

MTBF	MTTR	4	5	20	100
hours		year	year	year	year
50,000	30	8.45E-09	1.06E-08	4.23E-08	2.12E-07
10^5	30	5.29E-10	6.61E-10	2.65E-09	1.32E-08
10^6	30	5.28E-14	6.63E-14	2.65E-13	1.33E-12
50,000	100	3.10E-07	3.88E-07	1.56E-06	7.78E-06
10^5	100	1.94E-08	2.43E-08	9.77E-08	4.89E-07
10^6	100	1.95E-12	2.44E-12	9.80E-12	4.91E-11

3.2. Mirrored Layouts

Mirroring is functionally the simplest way to induce redundancy. Two copies are written, but either copy can satisfy reads. We can restore its contents by simply accessing the other copy.

3.2.1. The $3+3$ **Mirrored Layout** We present the $3+3$ mirrored layout in Figure 9. The array can tolerate any loss of a single drive and definitely any loss of four drives leads to data loss. However, losing two drives containing the same data leads to data loss. This happens with probability $\frac{1}{5}$ after loss of a single drive (there are 5 drives left and loss of the one containing the same data as the already failed drive leads to data loss.) If the array has tolerated two failures without data loss, it is (modulo renaming of disks) in the situation depicted in Figure 9. The chance that an additional loss looses access to the data in A or B is $\frac{1}{2}$.

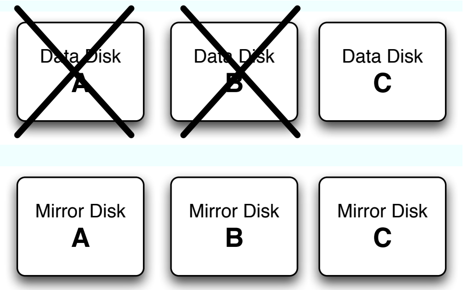

Figure 9. $3+3$ *Mirrored Layout surviving the loss of two drives.*

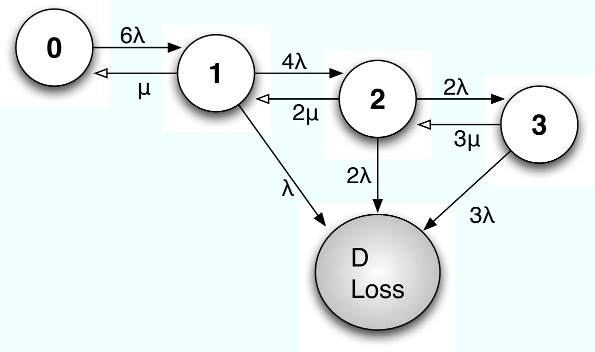

Figure 10. *Markov model for the* $3+3$ *mirrored layout.*

As a result, we obtain the Markov model depicted in Figure 10. We calculate the four and five year data loss rate of such a system and present the results in Table 4.

Table 4. Data loss probability of the mirrored array with six disks.

MTBF	MTTR	4	5	20	100
hours		year	year	year	year
50,000	30	2.51E-03	3.14E-03	1.25E-02	6.11E-02
10^5	30	6.30E-04	7.87E-04	3.15E-03	1.56E-02
10^6	30	6.31E-06	7.88E-06	3.15E-05	1.58E-04
50,000	100	8.31E-03	1.04E-02	4.09E-02	1.89E-01
10^5	100	2.09E-03	2.61E-03	1.04E-02	5.11E-02
10^6	100	2.10E-05	2.62E-05	1.05E-04	5.26E-04

3.2.2. The $4+4$ **Mirrored Layout** The derivation of the Markov model and the Kolmogorov system proceeds in strict analogy to the case of 6 disks. We give the Markov model in Figure 11 and the data loss probability during the economic lifespan of the array in Table 6.

3.3. Comparative Results

Figure 12 illustrates the relative likelihood of data loss for mean time between failure (MTBF) values ranging from

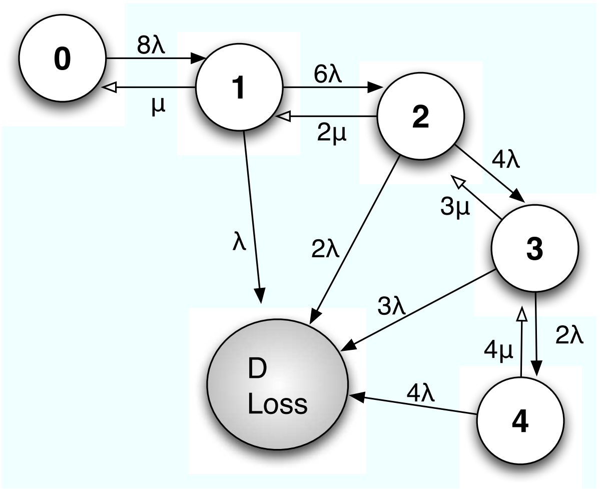

Figure 11. *Markov model for the mirrored array with eight disks.*

Table 5. Data loss probability of the mirrored array with eight disks.

MTBF	MTTR	4	5	20	100
hours		year	year	year	year
50,000	30	3.35E-03	4.19E-03	1.67E-02	8.06E-02
10^5	30	8.40E-04	1.05E-03	4.19E-03	2.08E-02
10^6	30	8.41E-06	1.05E-05	4.21E-05	2.10E-04
50,000	100	1.11E-02	1.38E-02	5.42E-02	2.43E-01
10^5	100	2.78E-03	3.48E-03	1.39E-02	6.75E-02
10^6	100	2.80E-05	3.50E-05	1.40E-04	7.01E-04

fifty thousand hours to a million hours, and an expected repair time of one hundred hours. Lower values therefore indicate a more reliable system. For each expected MTBF value we plot the expected likelihood of data loss for each layout during a four, five, twenty, and hundred-year period. All the layouts discussed in Section 3 were chosen to have identical storage capacity overhead, but varied in the number of nodes among which data was distributed and in the case of SSPiRAL we have also varied the number of nodes participating in a parity computation – the *x*-order. Figure 13 shows the same comparison of analytical results, but under an assumption of a shorter repair time (thirty hours). As expected, reducing the number of nodes holding data from four to three results in a decrease in the likelihood of data loss, but such a decrease was dwarfed by the impact of variations in MTBF, repair time, and the *x*-order.

The use of a fixed *x*-order for the SSPiRAL layouts implies that the data written to *x* data nodes must be combined and a parity computed to be written to one or more of the parity nodes. The additional computational effort may be minor, but poorly managed these parity calculations can pose a serious performance bottleneck. The primary cause of such a slowdown would be the potential of

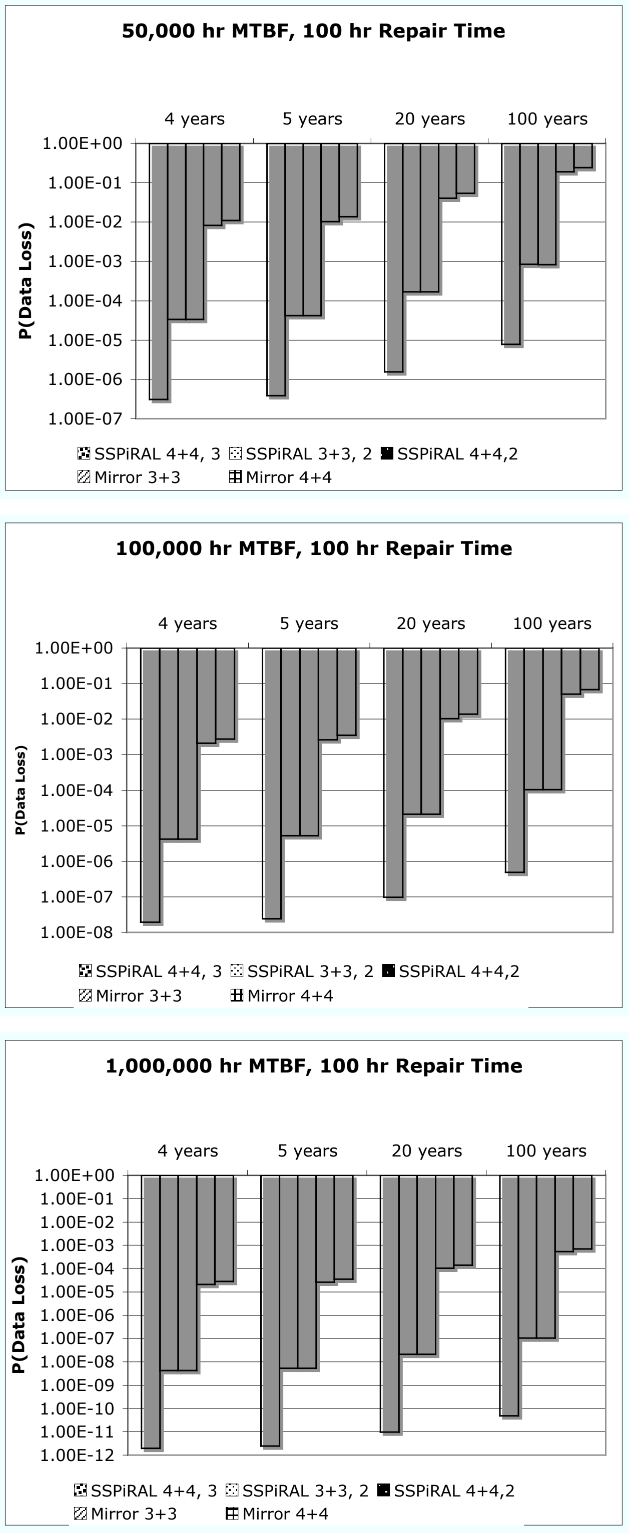

Figure 12. *Data loss probability for SSPiRAL and Mirrored layouts with equivalent space efficiency and a 100 hour repair time.*

parity updates overwhelming a parity disk. For example, four small random updates to four independent blocks on the data devices, while parallelizable across those same devices, would result in a sequential set of four update requests to the device holding the parity. While this problem can be mitigated by techniques such as declustering the data and parity, our preferred approach is to stripe a larger data block across all data devices. This avoids both the problem of the parity device being a bottleneck, and the further problem of updates to individual devices forcing an access to the parity device. When implemented, a large block update becomes an operation that always results in a single access to all devices involved. So while SSPiRAL implementations would have a performance impact compared to straightforward mirroring, it can be largely mitigated. The benefit of employing a SSPiRAL scheme becomes apparent when we look note from Figure 12 that the worst of the evaluated SSPiRAL schemes offers more than one hundred times less likelihood of failure than the best of the equivalent mirrored schemes. It is also interesting to see how increasing the *x*-order, from a simple pairwise parity, to a three-way parity, reduces the likelihood of failure by a further hundred times. These differences become more pronounced and impressive as the period of interest is increased. While the likelihood of data loss over a hundred years may seem like an excessive observation period for a particular storage system, it is not an unreasonable expectation when considering archival applications. Over a period of a hundred years, and assuming a fifty thousand hour MTBF, the likelihood of data loss for the SSPiRAL layouts is almost ten thousand times less than the equivalent mirrored schemes.

As the MTBF increases, whether due to the use of more reliable hardware, or building upon nodes that are themselves fault-tolerant, the overall likelihoods of data loss are reduced, but we see an even more pronounced difference between the SSPiRAL and mirrored schemes. At its most extreme, the SSPiRAL layouts are over a million times less likely to suffer data loss. But as repair times are decreased, we see this difference being much less pronounced. While different real-world scenarios would result in different repair times, the SSPiRAL layouts can be considered less affected by increases in these repair times than their mirrored counterparts.

From Figure 12 and Figure 13 we can see that the lowest likelihoods of data loss are achieved by the layouts with the largest number of participants in a parity computation. The *x*-order of a SSPiRAL layout has a greater impact on reducing the likelihood of data loss than decreasing the number of nodes. As the MTBF is increased, *e.g.,* by using more reliable sub-components or building larger storage farms from redundant disk arrays, the impact of employ-

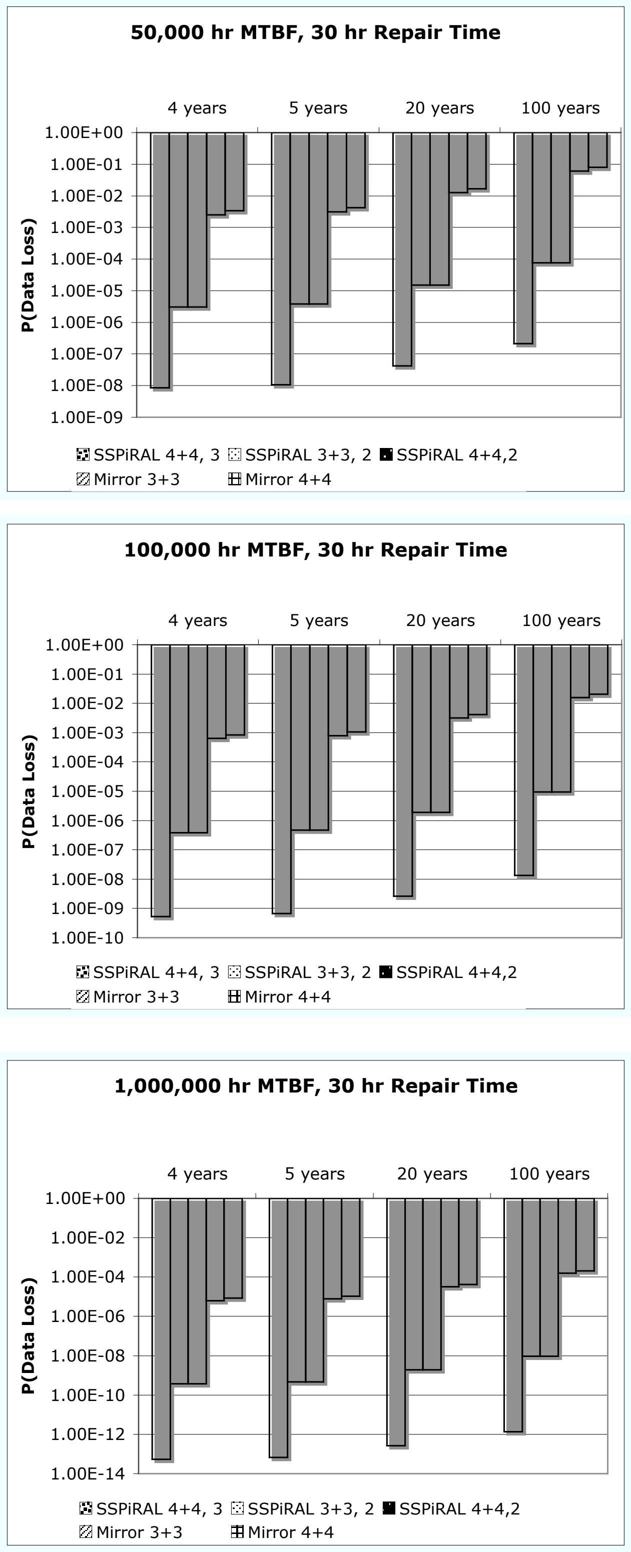

Figure 13. *Data loss probability for SSPiRAL and Mirrored layouts with equivalent space efficiency and a 30 hour repair time.*

ing SSPiRAL over simple mirroring schemes is even more pronounced.

4. Related Work

Like most of the original RAID layouts [5, 19], SSPiRAL is based solely on parity computations, and like more recent efforts [1, 3, 4, 10] SSPiRAL aims to survive the failure of multiple disks, and to achieve this goal efficiently. SSPiRAL diverges from prior efforts in its definition of efficiency. Unlike row-diagonal parity [4], SSPiRAL does not pursue the goal of optimizing capacity usage, and yet maintains the goals of optimal computational overhead and ease of management and extensibility. SSPiRAL replaces the goal of surviving a *specific* number of disk failures with the goal of surviving the most disk failures possible within the given resource constraints. The basic SSPiRAL layout discussed above can be described as an application of Systematic codes [20] across distinct storage devices. Similarly, such basic SSPiRAL layouts, in their limiting of the number of data sources, are similar to the fixed *in-degree* and *out-degree* parameters in Weaver codes [7] and the earlier $\hat{\mathrm{B}}$ layouts [22]. Weaver and $\hat{\mathrm{B}}$ are the most similar schemes to SSPiRAL, and all are parity-based schemes using principles first applied in erasure codes for communications applications such as the Luby LT codes, and the later Tornado and Raptor variants [14, 15, 21]. These codes all belong to the class of erasure codes known as low-density parity-check (LDPC) codes. They distinguish themselves from earlier Reed-Solomon and IDA codes by being more efficient to compute at the expense of space utilization. SSPiRAL differs from these prior applications of erasure codes in two major respects: it promises to be more efficient to maintain, and it is implemented with a direct consideration of available system resources, and departing from the requirement to tolerate only a fixed number of device failures. More closely related is the work of Hafner and Rao [8] that spoke to the MTTDL of non-MDS erasure codes (which would include codes such as SSPiRAL), and the scheme for surviving multiple drive failures patented by Wilner which is similar to the pairwise parity (*x*-order 2) SSPiRAL layouts [23].

5. Conclusion

We have presented the first complete evaluation of the reliability of SSPiRAL storage arrays consisting of *n* data disks and *n* parity disks where each parity disk contains the exclusive or (XOR) of two or three of the n data disks. While we assumed a simple failure rate for devices, and a more precise result might be obtained with a more complex and failure distribution, our analysis avoided the inaccuracies introduced by overly simplistic MTTDL results. Unlike previous studies, our results reflect the actual evolution of each storage array over its actual lifetime. Our results indicate that the three SSPiRAL schemes we considered were much more reliable than mirrored disk arrays with the same overhead. For instance, our $3+3$ SSPiRAL organization was found to be 10,000 times less likely to fail than a mirrored array consisting of three pairs of disks. We obtained even better results with a $4+4$ SPPiRAL organization where each parity disk contains the XOR of three of the four data disks as it was found to be one million times less likely to fail than a mirrored array consisting of four pairs of disks.

Directions for future work include investigating more complex SSPiRAL schemes, including schemes with variable *x*-order, and letting SSPiRAL arrays react to disk failures by dynamically reorganizing themselves while awaiting the replacement of the failed disk(s). When applied to arrays of mirrored disks, the technique was found to provide significant increases in system reliability while tolerating longer disk repair times [18].

One possible extension of this work is to consider alternative device failure models, and in particular to abandon the assumption of independent device failures. While this assumption simplified the analytical model, it did not favor SSPiRAL over mirroring, as we have been very conservative in our comparison. It is possible to further improve the estimated reliability of a SSPiRAL layout if we were to exploit knowledge of the expected failure rates of individual devices. Such expected values need not be precise and can be based on SMART information, the age of a device, the rate and number of power cycles it has experienced, or the state of other devices from the same manufacturer or production batch. The ability to classify devices into two or more classes based on their relative likelihood of failure would allow SSPiRAL layouts to assign the more failure-prone devices to less critical nodes, thereby improving the overall reliability of the layout compared to schemes that assume homogeneity and independence among devices. In this paper we have assumed no such advantage for SSPiRAL layouts. Further future work can include the exploitation of heterogeneity in SSPiRAL layouts for purposes beyond simply improving reliability. With the availability of multiple paths to reconstruct desired data, it is possible to use SSPiRAL layouts as a means to reduce overall disk activity, thereby offering opportunities to improve bandwidth, load balancing, or power savings. These goals potentially conflict, and the best mechanism to achieve them is still under investigation.

6. Acknowledgments

This work was supported in part by the National Science Foundation under Award no. 0720578, by the Department of Energy Office of Science under contract no. DE-FC02-06ER25768 and by the sponsors of the Storage Systems Research Center.

References

[1] G. A. Alvarez, W. A. Burkhard, and F. Cristian. Tolerating multiple failures in RAID architectures with optimal storage and uniform declustering. In *Proceedings of the 24th International Symposium on Computer Architecture (ISCA)*, pages 62–72, Denver, CO, USA, 1997. ACM.

[2] A. Amer, J.-F. Paris, T. Schwarz, V. Ciotola, and J. Larkby-Lahet. Outshining Mirrors: MTTDL of Fixed-Order SSPiRAL Layouts. In *Proceedings of the International Workshop on Storage Network Architecture and Parallel I/Os (SNAPI07)*, San Diego, CA, USA, Sept. 2007.

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

[4] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar. Row-diagonal parity for double disk failure correction. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, pages 1–14, San Francisco, CA, USA, 2004. USENIX Association.

[5] G. A. Gibson. *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*. PhD thesis, University of California at Berkeley, 1990.

[6] H. Gropp. *The CRC Handbook of Combinatorial Designs*, chapter IV.6 Configuration. CRC Press, 1996.

[7] J. L. Hafner. Weaver codes: Highly fault tolerant erasure codes for storage systems. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, San Francisco, CA, USA, Dec. 2005.

[8] J. L. Hafner and K. Rao. Notes on reliability models for non-MDS erasure codes. Technical Report RJ10391 (A0610-035), IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, Oct. 2006.

[9] L. Hellerstein, G. Gibson, R. Karp, R. Katz, and D. Patterson. Coding techniques for handling failures in large disk arrays. *Algorithmica*, 12(2-3):182–208, 1992.

[10] K. Hwang, H. Jin, and R. Ho. RAID-x: A new distributed disk array for I/O-centric cluster computing. In *Proceedings of the 9th IEEE Internation High Performance Distributed Computing Symposium (HPDC)*, pages 279–286, 2000.

[11] Z. Jie, W. Gang, L. Xiaogugang, and L. Jing. The study of graph decompositions and placement of parity and data to tolerate two failures in disk arrays: Conditions and existence. *Chinese Journal of Computers*, 26(10):1379–1386, Oct. 2003.

[12] W. Litwin, R. Moussa, and T. Schwarz. LH*RS – a highly-available scalable distributed data structure. *Transactions on Database Systems (TODS)*, 30(3), Sept. 2005.

[13] D. D. E. Long, B. R. Montague, and L.-F. Cabrera. Swift/RAID: A distributed RAID system. *Computing Systems*, 7(3):333–359, 1994.

[14] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, 2001.

[15] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC)*, pages 150–159, New York, NY, USA, 1997. ACM Press.

[16] P. Lyman and H. R. Varian. How much storage is enough? *ACM Queue*, 4(4), June 2003.

[17] P. Lyman and H. R. Varian. How much information?, Mar. 2007. http://www.sims.berkeley.edu/how-much-info-2003.

[18] J.-F. Pâris, T. J. E. Schwarz, and D. D. E. Long. Self-adaptive disk arrays. In *Proceedings of the 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 469–483, Dallas, TX, USA, Nov. 2006.

[19] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of SIGMOD*, pages 109–116. ACM, 1988.

[20] J. S. Plank and M. G. Thomason. A practical analysis of low-density parity-check erasure codes for wide-area storage applications. In *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Florence, Italy, June 2004.

[21] A. Shokrollahi. Raptor codes. *IEEE/ACM Transactions on Networking*, 14(SI):2551–2567, 2006.

[22] B. T. Theodorides and W. A. Burkhard. B̂: Disk array data layout tolerating multiple failures. In *Proceedings of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 21–32, Monterey, CA, USA, 2006.

[23] A. Wilner. Multiple drive failure tolerant RAID system. US Patent US 6,327,672 B1, LSI Logic Corporation, Milpitas, CA, Dec. 2001.