Using Shared Parity Disks to Improve the Reliability of RAID Arrays

Jehan-François Pâris
Department of Computer Science
University of Houston
Houston, TX 77204-3010
paris@cs.uh.edu

Ahmed Amer
Department of Computer Engineering
Santa Clara University
Santa Clara, CA 95050
a.amer@acm.org

***Abstract*—We propose to increase the reliability of RAID level 5 arrays used for storing archival data. First, we identify groups of two or three identical RAID arrays. Second, we add to each group a shared parity disk containing the diagonal parities of their arrays. We show that the new organization can tolerate all double disk failures and between 75 and 89 percent of triple disk failures without incurring any data loss. As a result, the additional parity disk increases the mean time to data loss of the arrays in the group it protects by at least 14,000 percent.**

Keywords- disk arrays, RAID arrays, fault-tolerance, storage system reliability.

I. INTRODUCTION

Archival storage systems differ in two important ways from conventional storage systems. First, they have to guarantee the integrity of their data over much longer periods of time, which often measure in decades. Second, these data are not supposed to be altered once they are stored in the system. As a result, archival storage systems tend to have much higher reliability requirements than most conventional storage systems. At the same time, the immutable nature of the data opens new avenues for the design of fault-tolerant storage architectures since the cost of update operations ceases to be an important consideration.

Wildani et al. [32] recently proposed a novel redundancy scheme for archival stores. They partition each disk into fixed-size "disklets" that are used to form conventional RAID stripes. In addition, they group these stripes into larger units, called "supergroups," and add to each supergroup one or more distinct "superparity" devices. The main advantage of the scheme is the higher reliability it provides, as superparity devices can participate in the recovery of reliability stripes that cannot recover on their own.

We propose here a streamlined variant of that concept. First, we do not partition disks and use instead conventional RAID arrays [7, 9, 20]. Second, we group these arrays into small groups of two to three identical arrays. Finally, we use a single superparity device to complement the parity blocks of the arrays in the group. The outcome of this process is an organization that can tolerate all double disk failures and between 75 and 89 percent of triple disk failures without incurring any data loss. As we will see, this improved reliability results in an increase of at least 14,000 percent of the mean time to data loss of the array pair.

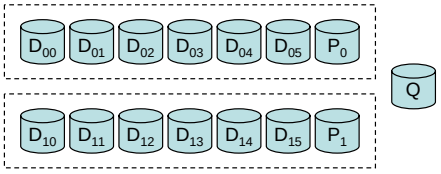

Figure 1. A pair of RAID arrays with an shared parity disk.

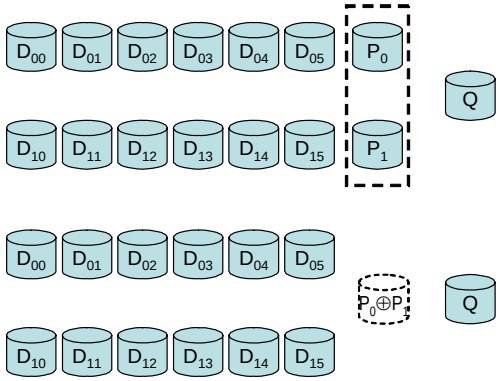

Figure 2. An alternate view of the previous array.

The remainder of this paper is organized as follows. Section II introduces our technique. Section III evaluates the reliability of pairs of arrays with an extra parity disk. Section IV discusses some implementation issues and Section V reviews previous work. Finally Section VI has our conclusions.

II. OUR TECHNIQUE

Consider the disk array displayed in Fig. 1. It consists of two conventional RAID arrays sharing an additional parity disk Q. For the sake of simplicity, we have represented the two arrays as having separate parity disks while we expect their parity blocks to be distributed among the seven disks forming each RAID array.

As Fig. 2 shows, we can define a *virtual parity disk P'* whose contents are the *exclusive or* (XOR) of the contents of parity disks P_0 and P_1. (Had the parity blocks been

equally distributed among the seven disks of each array, we would have defined a *virtual set of parity blocks*.)

Consider now the virtual array formed by the 12 data disks and the virtual parity disk *P'*. It forms a conventional RAID array that protects its contents against any single disk failure. We propose to use parity disk Q as an additional parity disk to protect the array against two simultaneous disk failures. This can be done by using an EvenOdd scheme [3], a Row Diagonal Parity (RDP) scheme [5,11] or any other RAID level 6 organization.

Going back to our original organization, we observe that the two parity disks P_0 and P_1 effectively protect all stored data against any single disk failure. As we have just seen, the three parity disks P_0, P_1 and Q also protect the same data against any double disk failure. Let us show now that they also protect the array data against most, but not all, triple disk failures.

We observe that our organization will be able to tolerate the failure of:

1. one arbitrary disk in each RAID array plus the shared parity disk Q as each array will be able to recover any lost data;
2. two arbitrary disks in a RAID array plus one arbitrary disk in the other array: the recovery process will be more complicated as we need to recover first any lost data in the second array before handling the double failure in the first disk array using the shared parity disk Q.

As seen in Fig. 3, the only triple failures that will result in a data loss are the failures of:

1. three disks in the same RAID array, or
2. two disks in the same RAID array *plus* the shared parity disk Q.

Since our disk organization comprises 15 disks, it can experience $\binom{15}{3}$ distinct triple failures. In addition, there are $\binom{7}{2}$ distinct double and $\binom{7}{3}$ distinct triple failures for each of the two RAID arrays. As a result, our disk organization will be able to tolerate exactly $\binom{15}{3}-2\binom{7}{3}-2\binom{7}{2}=343$ of the 455 possible triple disk failures, that is, slightly more than 75% of them.

More generally, we start with m RAID arrays comprising n disks each. We add to these mn disks an additional shared parity disk Q. We define a virtual parity disk *P'* that is formed by XORing the parity blocks of the m RAID arrays and form a single RAID level 6 array with the data blocks of the original arrays, the virtual parity disk *P'* and the shared parity disk Q. As our disk organization comprises $mn+1$ disks, it is subject to $\binom{mn+1}{3}$ distinct triple failures. Since there are $\binom{n}{2}$ distinct double and

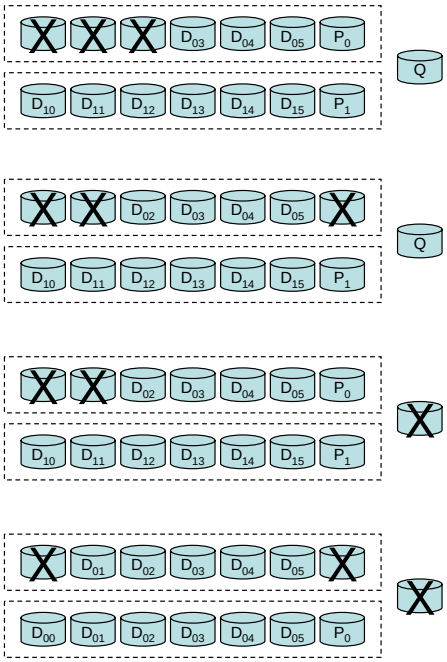

Figure 3. Triple failures resulting in a data loss.

$\binom{n}{3}$ distinct triple failures for each of the m RAID arrays, our disk organization will be able to tolerate $\binom{mn+1}{3}-m\binom{n}{3}-m\binom{n}{2}$ of the $\binom{2n+1}{3}$ possible triple disk failures, which happens to be 75% of them for two groups of disks (m = 2) and around 89% of them for three groups of disks (m = 3).

III. Performance Evaluation

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. When 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 are exponentially distributed with mean λ. In addition, we require repairs to be exponentially distributed with mean μ. Both hypotheses are necessary to represent each system by a Markov process with a finite number of states.

Building an accurate state-transition diagram for our disk organization is a daunting task as we must distinguish

between failures of the shared parity disk Q and failures of the other disks as well as between failures of disks belonging to the same disk array and failures of disks belonging to distinct arrays. Instead, we present here a simplified model.

Since disk failures are independent events exponentially distributed with rate λ, the rate at which an array that already has two failed disks will experience a third disk failure is $(15 - 2)\lambda = 13\lambda$. Observing there are 455 possible configurations with 3 failed disks out of 15 but 343 of them do not result in a data loss, we will assume that the rate at which a system that has two failed disks will experience a data loss will be $(545 - 343)\times 13\lambda/455 = 1456\lambda/455$.

Fig. 4 displays the simplified state transition probability diagram for a pair of RAID arrays with seven disks each and a shared parity disk Q. State <0> represents the normal state of the system when its 15 disks are all operational. A failure of any of these disks would bring the system to state <1>. A failure of a second disk would bring the array into state <2>. A failure of a third disk will either result in a data loss or bring the array to state <3>. Any fourth failure occurring while the array is in state <3> will necessarily result in a data loss.

Repair transitions return the array from state <3> to state <2> then from state <2> to state <1> and, finally, from state <1> to state <0>. Their rates are equal to the number of failed disks times the disk repair rate μ.

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

$$\frac{dp_0(t)}{dt} = -15\lambda p_0(t) + \mu p_1(t)$$
$$\frac{dp_1(t)}{dt} = -(14\lambda + \mu) p_1(t) + 15\lambda p_0(t) + 2\mu p_2(t)$$
$$\frac{dp_2(t)}{dt} = -(13\lambda + 2\mu) p_2(t) + 14\lambda p_1(t)$$
$$\frac{dp_3(t)}{dt} = -(12\lambda + 3\mu) p_3(t) + \frac{343}{455} 13\lambda p_2(t)$$

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

$$sp_0^*(s) - 1 = -15\lambda p_0^*(s) + \mu p_1^*(s)$$
$$sp_1^*(s) = -(14\lambda + \mu) p_1^*(s) + 15\lambda p_0^*(s) + 2\mu p_2^*(s)$$
$$sp_2^*(s) = -(13\lambda + 2\mu) p_2^*(s) + 14\lambda p_1^*(s) + 3\mu p_3^*(s)$$
$$sp_3^*(s) = -(12\lambda + 3\mu) p_3^*(s) + \frac{343}{455} 13\lambda p_2^*(s)$$

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

$$MTTDL = \sum_{i=0}^{3} p_i^*(0) ,$$

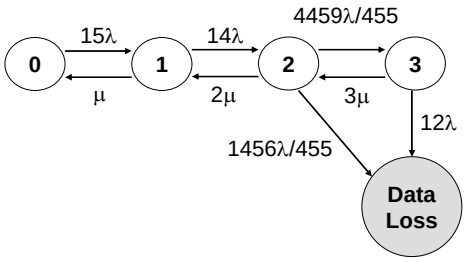

Figure 4. Simplified state transition probability diagram for a pair of RAID arrays with seven disks each and a shared parity disk.

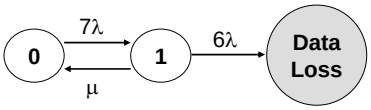

Figure 5. State transition probability diagram for a RAID array with seven disks.

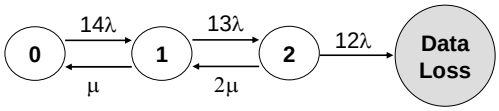

Figure 6. State transition probability diagram for a RAID level 6 array with 14 disks.

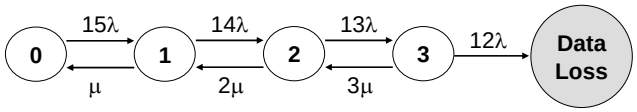

Figure 7. State transition probability diagram for a 12-out-of-15 array with 15 disks.

we solve the system of Laplace transforms for $s = 0$ and use this result to compute the MTTDL of our system

$$MTTDL = \frac{7585\lambda^3 + 1187\lambda^2\mu + 103\lambda\mu^2 + 5\mu^3}{420\lambda^3(65\lambda + 4\mu)} .$$

Our performance study would not be complete if we did not compare that MTTDL against that of comparable disk arrays. The three benchmarks we selected were:

1. a pair of RAID arrays with seven disks each,
2. a single RAID level 6 array with 14 disks,
3. a 12-out-of-15 disk array tolerating three disk failures.

All three arrays have the same storage capacities as our system, that is, 12 data disks.

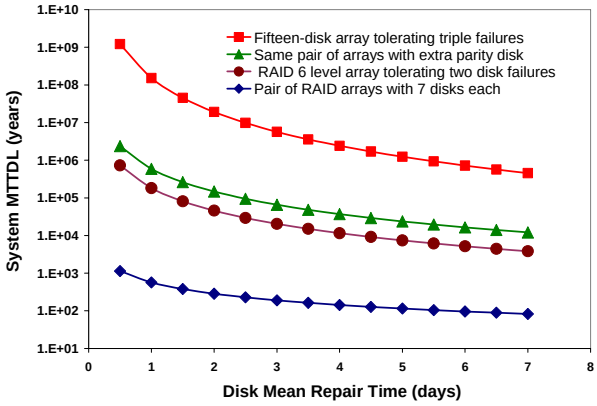

Figure 8. Compared MTTDLs of the four array organizations.

Deriving the MTTDLs of these three organizations was a fairly easy task because the MTTDL of a pair of RAID arrays is one half of that of a single RAID array and the two other organizations are both *m*-out-of-*n* disk arrays. Figures 5 to 7 display their respective state transition probability diagrams. In all three diagrams, state <0> represents the initial state of the array when all disks are operational.

Figure 8 displays on a logarithmic scale the MTTDLs achieved by our proposed organization and compares them with the MTTDLs achieved by its three benchmarks. We assumed that the disk failure rate λ was one failure every one hundred thousand hours, that is, slightly less than one failure every eleven years. These values correspond to the high end of the failure rates observed by Pinheiro et al. [25] and Schroeder and Gibson [27]. Disk repair times are expressed in days and MTTDLs in years.

As we can see, adding a shared parity disk to the two RAID arrays increases their MTTDL by at least 14,000 and up to 20,000 percent. Our new organization also bests a RAID level 6 organization with 14 disks by more than 200%. At the same time, it performs significantly worse than the 12-out-of-15 array.

These results were to be expected. Recall that:

1. A pair of RAID arrays can tolerate all single disk failures and some double disk failures, but not those involving two disks in the same array.
2. A single RAID level 6 array can tolerate all single and double disk failures but no triple disk failures.
3. A pair of RAID arrays with a shared parity disk can tolerate all single or double disk failures and 75 percent of triple disk failures.
4. A 12-out-of-15 disk array can tolerate all single, double or triple disk failures.

A more unexpected observation is the impact of fatal triple failures on the MTTDL of our organization. Even though our scheme tolerates 75 percent of all triple failures, it does not perform as well as the organization tolerating all triple failures.

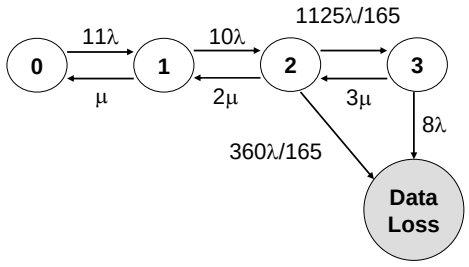

Figure 9. Simplified state transition probability diagram for a pair of RAID arrays with five disks each and a shared parity disk.

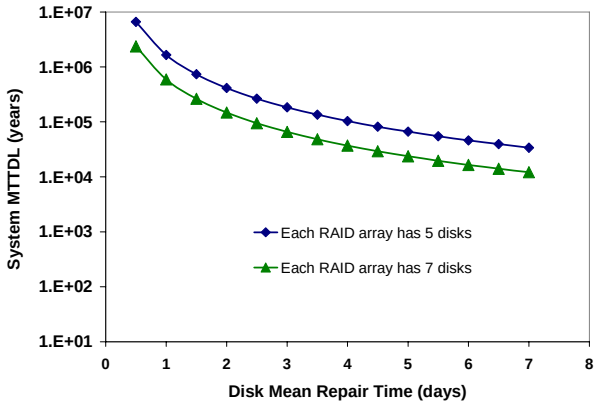

Figure 10. Compared MTTDLs of pairs of RAID arrays with an extra parity disk.

We also evaluated a smaller organization consisting two RAID arrays with five disks each and an additional shared parity drive. Since this smaller organization comprises 11 disks, it can experience $\binom{11}{3}$ distinct triple failures. As there are $\binom{5}{2}$ distinct double and $\binom{5}{3}$ distinct triple failures for each of the two RAID arrays, the smaller disk organization will be able to tolerate $\binom{11}{3}-2\binom{5}{3}-2\binom{5}{2}$ of the $\binom{11}{3}$ possible triple disk failures, that is, 125 out of 165.

Fig. 9 shows the simplified state transition probability diagram for the smaller organization. Observe that the failure rates were adjusted to reflect the smaller number of disks. In addition, the rate at which an array that has two failed disks will experience a data loss is now $(165 - 125)\times 9\lambda/165 = 360\lambda/165$.

Using the same techniques as before, we obtain the MTTDL of the smaller organization

$$MTTDL = \frac{17281\lambda^3 + 3935\lambda^2\mu + 487\lambda\mu^2 + 33\mu^3}{3960\lambda^3(11\lambda + \mu)}.$$

Fig. 10 displays on a logarithmic scale the MTTDLs achieved by our two proposed organizations. As we can see, the smaller organization achieves MTTDLs that are 180 percent higher than those achieved by the larger organization. This should not surprise us because the smaller organization uses three parity disks (or their equivalents) to protect the contents of eight data disks while the larger organization uses the same number of parity disks to protect the contents of twelve data disks. As a result, the smaller organization has a larger redundancy level than the larger one, which translates into a higher MTTDL.

Finally, let us consider the case of a group of three RAID arrays with five disks each and an additional shared parity disk. As seen on Fig. 11, this new organization comprises 16 disks. As there are $\binom{5}{2}$ distinct double and $\binom{5}{3}$ distinct triple failures for each of the three RAID arrays, our disk organization will be able to tolerate $\binom{16}{3}-3\binom{5}{3}-3\binom{5}{2}$ of the $\binom{16}{3}$ possible triple disk failures, that is, 500 out of 560 or about 89 percent.

Fig. 12 shows the simplified state transition probability diagram for our third organization. Observe that the rate at which an array that has two failed disks will experience a data loss is (560 – 500)×12λ/560 =840λ/500.

Using the same techniques as before, we obtain the MTTDL of our organization

$$MTTDL = \frac{23524\lambda^3 + 2915\lambda^2\mu + 253\lambda\mu^2 + 12\mu^3}{240\lambda^3(364\lambda + 9\mu)}.$$

Note that our model assumes that all quadruple failures will result in a data loss. This is not true as our disk organization can tolerate some quadruple losses such as the failures of one disk in each RAID array plus the shared parity disk or the failures of two disks in a RAID array plus one disk in each of the two other arrays. We can safely neglect the contributions of these configurations to the MTTDL of the array as long as the disk repair rate μ is much higher than the disk failure rate λ because the probability that our disk organization has four failed disks is then negligible compared to the probability that it has three failed disks.

To estimate the benefits of adding a shared parity disk to a set of three RAID arrays, we also computed the MTTDL of the same arrays without the parity disk. Observing that the MTTDL of a set of three RAID arrays is one third of the MTTDL of an individual array we obtain

$$MTTDL = \frac{9\lambda + \mu}{60\lambda^2}$$

Fig. 13 displays on a logarithmic scale the MTTDLs achieved by the two organizations. As we can see, adding a single shared parity disk to the three RAID arrays increases their MTTDL by at least 18,000 and up to 277,000 percent. This latter number is not typical as it requires a repair time of half a day or less. Assuming a more typical disk replacement time of two days, we find that adding a shared parity disk to the three RAID arrays increases their MTTDL by slightly more than 68,000 percent.

There are three issues we should mention concerning the accuracy of our results. First, we assumed that failure occurrences and repair times followed exponential laws. This is not true for real disk populations since failures and

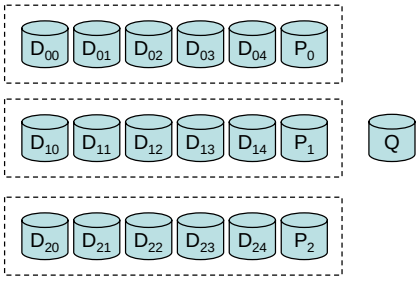

Figure 11. Three RAID arrays sharing an additional parity disk.

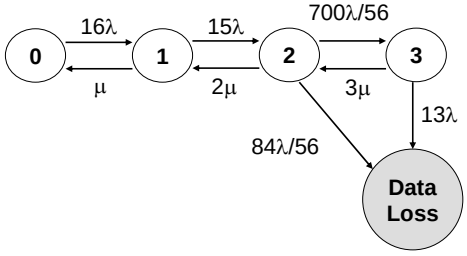

Figure 12. Simplified state transition probability diagram for three RAID arrays with five disks each sharing an extra parity disk.

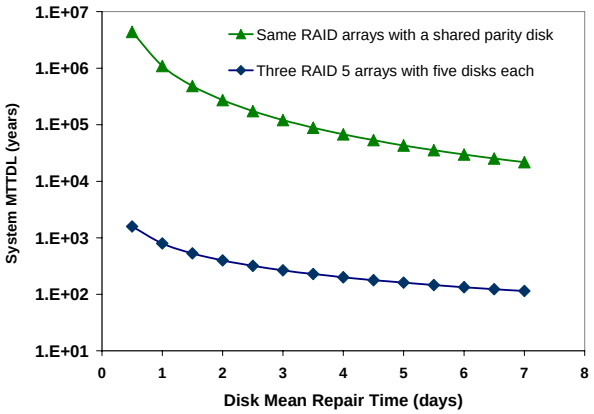

Fig. 13. Compared MTTDLs of three RAID arrays with five disks each with and without a shared parity disk

repair time distributions have much smaller coefficients of variation than the exponential distribution. Second, we assumed a constant failure rate λ over the lifetime of the array. In reality, disk failure rates tend to decrease over the first few months of the disk lifetime then remain constant for a few years and increase again as the disk wears out.

Finally, we used MTTDL to represent the reliability of disk arrays. MTTDLs characterize fairly well the behavior of disk arrays that would remain in service until they fail without being replaced for any reason other than a device failure. This is rarely the case as disk arrays are typically replaced after five to seven years, that is, well before they experience any failure. Since MTTDLs do not take into account this relatively short lifetime, they tend to overestimate the probability of a data loss over this lifetime. This effect remains negligible as long as the time to repair an individual disk is at least one thousand times shorter than its mean time to failure.

IV. Implementation Issues

The main limitation of our proposal is the heavy burden it puts on the shared parity disk *Q*, which has to be updated every time either of the two arrays is updated. We propose to discuss here some of the possible options.

The simplest solution to this problem is to restrict the application of our technique to archival data. Since these data are not typically altered once the archive has been created, the shared parity disk will have to handle few or no updates over the lifetime of the archive. One attractive option would be to rely on the protection offered by the RAID arrays while the archive is being created, and to postpone the additional protection offered by the shared parity disk until the archive has become stable.

Another option would be to increase the throughput for the shared parity disk. For instance, we could add to the disk a dedicated microprocessor and a sufficient amount of non-volatile storage to increase the disk bandwidth. A more attractive option would be to use storage class memories (SCMs). Unlike magnetic disks and MEMS [6], these new devices have no moving parts. In addition, they do not suffer from the write-speed limitations of flash memory. SCMs [19] are expected to have much faster access times than magnetic disks and data rates varying between 200 and 1,000 MB/s. In addition, their mean times to fail (MTTFs) are expected to exceed ten million hours and their write endurance to reach one billion write cycles.

Recall that the only triple failures that will result in a data loss are the failures of:

1. three data disks in the same RAID array, or
2. two disks in the same RAID array *plus* the shared parity disk *Q*.

This means that replacing the shared parity disk by a shared parity SCM would dramatically reduce the probability of observing a simultaneous failure of the shared parity device and two data disks in the same array. Having a much faster and much more reliable shared parity device would also allow us to let four or maybe more RAID arrays share a single SCM parity device, thus amortizing the higher cost of the new device over a larger number of protected disks.

An interesting consequence of this approach is that the failure of three disks in the same RAID array would become the dominant failure mode for our system. Excluding acts of God, such as fires, floods and thunderstorms, there are two main contributing factors for triple disk failures within a relatively small set of disks. First, the excessive heat produced by a failing disk can affect the neighboring disks. Second, we could observe a rapid succession of disk failures among disks belonging to the same production batch as a result of the manifestation of a hidden common defect in most, if not all, disk drives belonging to that batch.

Fortunately for us, there are easy defenses against these two risk factors. First, we should not place disks that belong to the same RAID array too close to each other. Second, we should try as much as possible to use disk drives belonging to different production batches in each RAID array.

V. Previous Work

Increases in data volumes inevitably result in larger numbers of devices, which in turn result in an increased likelihood of multi-device failures, and so there has been a significant amount of work on schemes to tolerate multi-device failures. Traditional RAID schemes aimed at surviving the loss of an individual device within an array [9, 20], and with variations of RAID-6 (and its various implementations) the goal was to survive the loss of two devices within an array [15, 21, 22]. EvenOdd and Row-Diagonal Parity are also parity-based schemes capable of surviving two-device failures [3, 5, 11]. The common goal of all these schemes was to survive the requisite number of device failures while attempting to minimize the total space sacrificed for redundant storage.

Other parity-based redundancy schemes included STAR, HoVer, GRID, and Bˆ, the latter of which typified the tendency of such approaches to focus on the data layout pattern, independent of the number of underlying devices [13, 14, 18, 29]. Typically, these data layouts were subsequently declustered data across homogenous, uniform, devices, and the majority could be classified as variations of low-density parity-codes similar to those used for erasure coding in the communications domain, such as the Luby LT codes, and their Tornado and Raptor variants [16, 17, 28]. Similar to the scheme we propose, HoVer and the more general GRID used parity-based layouts based on strips arranged in two or more dimensions. All these layouts all assumed uniform homogenous devices, and largely competed on their space efficiency [30, 33], or their ability to survive more than two device failures [14, 31].

Redundant layouts such as Bˆ, Weaver codes [12], and our own SSPiRAL schemes [1, 2, 8,] departed from this trend, and offered redundant layouts that strictly limited the number of devices contributing to parity calculations, thereby offering a practical scheme for greater numbers of

devices than those typically found in RAID arrays. SSPiRAL layouts were novel in their focus on individual device failures having potentially differing impact on the survivability of data. Self-repairing disk arrays constitute another worthwhile option: these arrays reorganize themselves whenever they experience a disk failure and return to their original configuration once the failed disks are replaced [23].

Wildani et al. [32] recently proposed a multi-level redundancy scheme for archival stores. They partition each disk into fixed-size "disklets" and use these disklets to form conventional RAID stripes. They group these stripes into larger units, called "supergroups," and add to each supergroup one or more "superparity" devices. As we mentioned earlier, the main advantage of the scheme is the higher reliability it provides, as superparity devices can participate in the recovery of reliability stripes that cannot recover on their own.

VI. Conclusion

We have presented a new disk array organization that increases the reliability of RAID level 5 arrays used for storing archival data by adding to small groups of RAID arrays an additional disk containing the diagonal parities of the arrays. We show that the new organization can tolerate all double disk failures and from 75 to 89 percent of triple disk failures without incurring any data loss. As a result, the additional parity disk increases the mean time to data loss of each array pair by at least 14,000 percent.

More work is still needed to evaluate the benefits of using solid state devices, such as storage-class memories for the shared parity disk.

References

[1] A. Amer, D.D.E. Long, J.-F. Pâris, and T. Schwarz, Increased reliability with SSPiRAL data layouts, *Proc. IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (MASCOTS '08), Baltimore, MD, pp. 89–198, Sep.2008.

[2] A. Amer, J.-F. Pâris, T. Schwarz, V. Ciotola, and J. Larkby-Lahet, Out-shining Mirrors: MTTDL of Fixed-Order SSPiRAL Layouts, *Proc. International Workshop on Storage Network Architecture and Parallel I/Os* (SNAPI '07), San Diego, CA, pp. 11–16, Sep 2007.

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

[4] W. A. Burkhard and J. Menon. Disk array storage system reliability. *Proc. 23rd International Symposium on Fault-Tolerant Computing*, Toulouse, France, pp. 432–441, June 1993.

[5] 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 2005) San Francisco, CA, pp. 1–14, 2004.

[6] L. R. Carley, G. R. Ganger, and D. F. Nagle, MEMS-based integrated-circuit mass-storage systems. *Communications of the ACM,* 43(11):73–80, Nov. 2000.

[7] 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.

[8] V. Ciotola, J. Larkby-Lahet, and A. Amer, SSPiRAL layouts: Practical extreme reliability, *Tech. Report TR-07-149*, Department of Computer Science, University of Pittsburgh, 2007, Presented at the Usenix Annual Technical Conference 2007 poster session.

[9] G.A. Gibson, Redundant disk arrays: Reliable, parallel secondary storage, *Ph.D. Thesis*, University of California, Berkeley, 1990.

[10] K.M. Greenan, E.L. Miller, and J.J. Wylie, Reliability of XOR-based erasure codes on heterogeneous devices, *Proc. Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN 2008), Anchorage, AK, pp. 147–156, June 2008.

[11] 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* (NCA 2008),Cambridge, MA, pp. 93–100, July 2008.

[12] J.L. Hafner, Weaver codes: Highly fault tolerant erasure codes for storage systems, *Proc. USENIX Conference on File and Storage Technologies* (FAST '05*)*, San Francisco, CA, Dec. 2005.

[13] J.L. Hafner, HoVer erasure codes for disk arrays, *Proc. International Conference on Dependable Systems and Networks* (DSN 2006), Philadelphia, PA, pp. 217–226, June 2006.

[14] 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 '05), San Francisco, CA, pp. 197–210, Dec. 2005.

[15] 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, 2003.

[16] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, and V. Stemann, Practical loss-resilient codes, *Proc. 29th ACM Symposium on Theory of Computing* (STOC '97), El Paso, TX, pp. 150–159, May 1997.

[17] 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.

[18] M. Li, J. Shu, and W. Zheng, GRID codes: Strip-based erasure codes with high fault tolerance for storage systems, *ACM Transactions on Storage,* 4(4):1–22, Jan. 2009.

[19] S. Narayan, Storage class memory a disruptive technology, *Presentation at Disruptive Technologies Panel: Memory Systems of SC '07*, Reno, NV, Nov. 2007.

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

[21] 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 '08), Cambridge. MA, pp. 85–92, July 2009.

[22] J. S. Plank, The RAID-6 liberation codes, *Proc. 6th USENIX Conference on File and Storage Technologies* 2 pp. 1–14, Feb. 2008.

[23] J.-F. Pâris, T. J. Schwarz and D. D. E. Long. Self-adaptive archival storage systems. *Proc. 26th International Performance of Computers and Communication Conference* (IPCCC '07), New Orleans, LA, pp. 246–253, Apr. 2007.

[24] J.S. Plank and M.G. Thomason, A practical analysis of low-density parity-check erasure codes for wide-area storage applications, *Proc. 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN '04), Florence, Italy, p. 115, June 2004.

[25] 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 '07), San Jose, CA, pp. 17–28, Feb. 2007.

[26] T. J. E. Schwarz and W. A. Burkhard. RAID organization and performance. *Proc. 12th International Conference on Distributed Computing Systems* (ICDCS ’92), Yokohama, Japan, pp. 318–325 June 1992.

[27] 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 '07), San Jose, CA, pp. 1–16, Feb. 2007.

[28] A. Shokrollahi, Raptor codes, *IEEE/ACM Trans. Networking* 52(6):2551–2567, June 2006.

[29] B.T. Theodorides and W.A. Burkhard, Bˆ: Disk array data layout tolerating multiple failures, *Proc. IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (MASCOTS '06), Monterey, CA, 2006, pp. 21–32, Sep. 2006.

[31] G. Wang, X. Liu, S. Lin, G. Xie, and J. Liu, Constructing double- and triple-erasure-correcting codes with high availability using mirroring and parity approaches, *Proc. 13th International Conference on Parallel and Distributed Systems* (ICPADS '07), Hsinchu, Taiwan, pp. 1–8, Dec. 2007.

[32] A. Wildani, T. J. E. Schwarz, 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 '09), London, GB, pp. 246–256, Sep. 2009.

[33] L. Xu and J. Bruck, X-code: MDS array codes with optimal encoding, *IEEE Trans. Information Theory* 45(1):272–276, 1999.