October 28, 1993

Multi-Dimensional Disk Array Reliability

Thomas J.E. Schwarz, S.J. [*]
Walter A. Burkhard [†]

Gemini Storage Systems Laboratory
University of California, San Diego
9500 Gilman Drive
La Jolla CA 92093-0114

Abstract

The excellent reliability provided by RAID Level 5 data organization has been seen to be insufficient for future mass storage systems. We analyze the multi-dimensional disk array in search of the necessary improved reliability. The paper begins by introducing multi-dimensional disk array data organization schemes based on maximum distance separable error correcting codes and incorporating both strings and spares. Several figures of merit are calculated using a standard Markov failure and repair model for these organizations. Based on our results, the multi-dimensional disk array organization is an excellent approach to providing improved reliability.

1 Introduction

Disk array storage systems, especially those with redundant arrays of independent disks (RAID) Level 5 data organization [6], provide excellent cost, run-time performance as well as reliability and will meet the needs of computing systems for the immediate future. Computing systems, especially those with massive storage requirements, may need even greater reliability than provided by RAID Level 5[1].

Our goal within this paper is to present and analyze data organization schemes for disk arrays that provides better reliabilities than RAID Level 5. The data organizations, derived from maximal

---

[*] Current address: St. Michael's Institute, Gonzaga University, 1107 North Astor Road, Spokane, Washington 99202

[†] Supported in part by grants from the University of California MICRO program as well as NCR Dayton, Ohio, NCR Rancho Bernardo, California and IBM Almaden Research Center, San Jose, California.

distance separable (MDS) codes, retain some of the performance advantages of RAID Level 5 while providing higher reliability[8]. Gibson et al. present the *multi-dimensional* parity schemes [3] which form the basis for the data organizations discussed here. Our data organizations incorporate novel combinations of spare disks [5] and strings[4].

Our paper is organized of follows. Section 2 contains a brief overview of the terminology we will use throughout. Within section 3 we present our failure assumptions and the Markov model used to analyze the reliability of these data organizations; we conclude this section with Table that contains typical failure and repair rates used within our calculations. In section 4 we consider four addressing schemes for one-dimensional disk arrays. Within section 5 we consider two-dimensional disk arrays; here we present five addressing schemes some of which are very similar to those of section 4. We conclude the section with a novel approach to providing high resiliency. Finally section 6 contains our conclusions where we review our results and discuss trade-offs for our various approaches.

2 Terminology

We begin by briefly reviewing the terminology we will be using throughout the paper.

A reliability group is a group of data storage elements, either whole disks or disk tracks, that are associated by shared data redundancy, such that failure of one (or more, depending on the scheme) element(s) leaves all the data accessible.

A RAID Level 5 data organization reliability group contains data disks and a single parity check disk [6]; an MDS data organization reliability group contains data disks and a pair of check disks[8]. Level 5 organizations can tolerate a single disk failure while the MDS organization withstands a pair of concurrent disk failures.

Our orthogonal organization of strings and reliability groups is similar to that of Gibson [3]. A *string* is a group of disks that share hardware components such as power supply and cabling, cooling, SCSI controller and cabling and host bus adapter(HBA)[4]. We consider three varieties of hardware redundancy within strings. A string is *soft* if it contains only the basic set of hardware components; as an example, a string with 10 disks would have two sets of HBAs and SCSI controllers, one power supply and one cooling fan. A string is *hardened* if some components are duplicated. Within this paper, our hardened strings will double the power supply, SCSI controllers and the HBAs components. A string is *super-hardened* if all the hardware components are duplicated. For us a super-hardened string will be a hardened string with duplicate cooling fans.

An *addressing* scheme is a means to improve RAID performance. They provide a translation between the physical disks layer and a logical layer. It can distribute check and spare space evenly through the disk array and thus avoids hot-spots during write bursts or after a disk failure. The addressing scheme is a reliability factor. For example an addressing scheme that first chooses a string and then a disk in the string to store a track never places tracks on the same physical string unless that is the case on the logical level.

3 Failure Modeling and Reliability

We consider three varieties of components within a disk array subsystem. There are (i) individual disk drives, (ii) essential components common to all disks and (iii) strings of disks. In our modelling, we will assume that strings can be either soft, hardened or super-hardened.

We do not model operation failure in this work [2] which can have a significant effect on reliability. However, its inclusion renders the analysis much more complex and we will use our current results to point in the proper direction when beginning such an effort.

For our reliability calculations, we assume that components fail with exponential (memoryless) probability. While this assumption ignores the well-known “bathtub” life expectancy of components and catastrophic failures, it makes the analysis tractable and provides meaningful reliability figures.

We model the disk array failure status with a continuous Markov model. The states describe the number of operational disks and strings and the transitions reflect component failure. State $\mathcal{F}$ designates data loss and is absorbing. The mean time to data loss (MTTDL) is the expected time to enter state $\mathcal{F}$. The reliability function $R(t)$ is the probability of not being in the failure state at time t and determines MTTDL by:

$$\text{MTTDL} = \int_0^\infty R(t)dt.$$

We can use the approximation

$$R(t) = e^{-t/\text{MTTDL}}$$

if we are able to determine the MTTDL. The state transitions are marginal probabilities which determine the individual state probabilities. The Laplace transform of the resulting differential state equations solves the integral expression for MTTDL quickly. If M is the matrix whose coefficients are the state transitions between non-failure states, then

$$\text{MTTDL} = -(1, 1, \ldots, 1) \cdot M^{-1} \cdot (1, 0, \ldots, 0)^T$$

where $(1, 0, \ldots, 0)^T$ designates the initial condition with all disks operational with probability one.

In Table 1 we summarize the transition event rates used in our calculations.

event		rates (per hour)	identifier
essential component failure		1×10^{-7}	ϵ
disk drive failure		2×10^{-5}	λ
string failure	soft	2×10^{-5}	μ
	hardened	5×10^{-6}	
	super-hardened	5×10^{-8}	
component repair		2.77×10^{-2}	ρ

Table 1: Event Rates

4 One-Dimensional Disk Array Organizations

We present four data organizations in this section that use level 5 and MDS reliability groups. Our orthogonal organization of strings and reliability groups is similar to that of Gibson [3].

The first two organizations considers different addressing modes for an ensemble of disks and we evaluate the MTTDL for each. More specifically, we assume n level 5 reliability groups each containing m disks. There are m strings of n disks and the m disks of each reliability group are distributed into the m different strings – one per string. The second pair of organizations utilizes these two addressing modes and extends them by distributing spare disks within the the ensemble.

Our first addressing scheme, designated 4A, is a two step procedure: 1) we select the target string and 2) we select the disk within the string. That is, within orthogonal array parlance, we select first the column and second the row. This scheme is able to withstand any single disk failure; however, data loss is virtually guaranteed with the failure of two disks not on the same string. This configuraton will not lose data with concurrent disk failures within a string or failure a string. This addressing scheme effectively distributes all data accesses "uniformly" over all disks thereby balancing the load. Our Markov model for the configuration is given in Figure 1. We can calculate the MTTDL within this model using parameters specified within Table 1. We use a continuing example throughout that contains 100 data disks. Thus for our first calculation, we have 11 strings each containing ten disks as shown in Figure 3. The 11 strings populate the eleven columns shown within the Figure. We have two host board adaptors as well as two scsi controllers (not shown) per string. The reliability group shown as a row within the array is only schematic for addressing mode 4A since step two requires us to determine a particular row within a column.

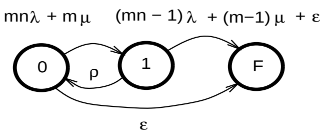

Figure 1: 3.A addressing

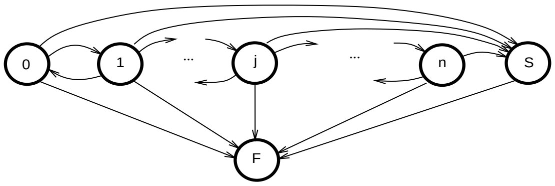

Figure 2: 3.B addressing

Markov Models for One Dimensional Disk Array: No Spares

Our second addressing scheme, designated 4B, is a single step procedure: we select a disk within a reliability group. This approach permutes the logical disks within the reliability group thereby moving the role of the check disk throughout the group. With this approach, data loss occurs if two disks in a reliability group are inaccessible. Our Markov model for this configuration is described in

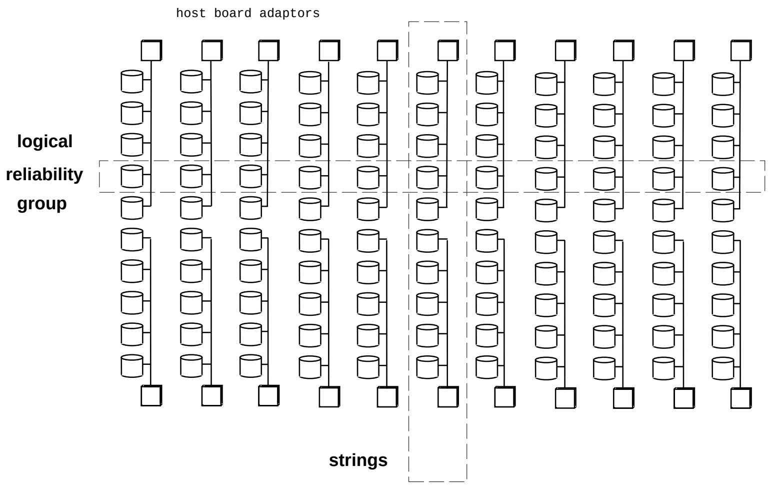

Figure 3: Disk Array Ensemble with Eleven Strings and Ten Reliability Groups

Figure 2; we have not described the transition rates here as the model is much larger. Our calculation however proceeds as for the previous model. The parameters are as before as is the configuration of the 100 data disks and the eleven strings. The reliability groups shown in Figure 3 are now both logical and physical; that is, our ten reliability groups correspond exactly to the rows of the array.

Much higher reliability figures are possible if spare disks are used. The third data organization, designated 4C, is an extension of 4A with the data being distributed throughout the array. The spare disks, residing on a single string, are not distributed. We augment our system configuration shown in Figure 3 to include an additional string of spare disks. Thus when a disk or string fails, we can reconstruct the unavailable data within the spare string. Rather than have a full string of n spares, we have only s spares. We can utilize the spares to store reconstructed data from a failed disk drive. With less than n spares, we cannot directly replace a failed string in this manner. However we use the spare disks as a repository for string data reconstructed on demand until the string is repaired. Our Markov model for this configuration is an extension of the model within Figure 1 with new states to represent the number of currently available spares and to characterize the failure of a string. Our model is optimistic since we assume the spares replace lost data immediately.

The fourth addressing scheme, designated 4D, has the best MTTDL figure within the paper.

disk array organization	MTTDL in years super strings	MTTDL in years hard strings	MTTDL in years soft strings
4A	62	59	13
4B	763	505	21
4C	4899	1822	32
4D	11537	11388	469
5A	11218	2177	29
5B	7254	1061	19
5C	7787	1072	19
5D	11147	1932	25
5E	11169	2005	27

Table 2: MTTDL Values

The scheme consists of a pair check disks per reliability group using an MDS organization. There are $m+2$ disks per reliability group. We assume the data is distributed only within the reliability groups as within 4B. This organization has the same reliability as a level 5 array having distributed sparing in which the spares are used for reconstruction only within the same reliability group. This data organization will suffer data loss exactly when at least three disks within a reliability group concurrently fail. Our Markov model for this configuration must characterize the number of groups with a single failure as well as the number with a pair. Some of these "failures" may be induced by a string failure. We present our results within Table 2.

5 Two-Dimensional Disk Array Organizations

The n^2-1 two-dimensional parity organization places each disk drive in two reliability groups to achieve excellent data protection. We also assemble the disks together in strings as demonstrated in Figure 4 that contains 24 disks on 6 strings each with 4 disks. Each reliability group has a RAID Level 3 or 4 organization. The disks of each string have different top patterns and the check disks are marked. The rightmost column and bottom row constitute strings; the remainder of the strings are "northwest-southeast" diagonals in the figure.

Our first addressing scheme, 5A, is a single step process simply selecting a (horizontal) reliability group. The group organization then determines the disk within the group. The role of all disks is fixed. We analyzed an organization of 120 disks grouped into 12 strings. We present the Markov model in a technical report [9] rather than here since it contains 33 states. The data survive all two disk failures, survive the failure of three disks almost always, and show higher susceptability to further disk failures; the data survive all single string failures as well as almost one third of the double string failures. The drawback to the scheme is bad write performance: A write burst within a group makes the check disks hot spots.

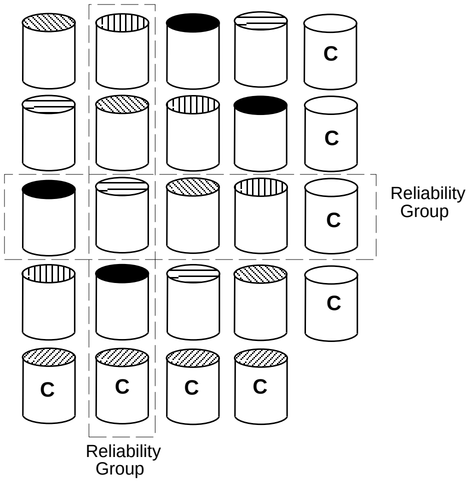

Figure 4: Two Dimensional Parity Disk Array

We improve the write performance by interchanging the role of disks. Our second organization, 5B, maps to any disk within the array. For a given track number, the logical disks are a permutation of the physical disks; this allows the check disk role to move among the entire array thereby avoiding the run-time performance problem mentioned previously. For example if a disk contains 1000 tracks, this addressing scheme imposes 1000 reliability groups on the physical disk array. Regretfully the impact on reliability is great. Now 30 per cent of all groups of three concurrent disk failures and virtually every failure of five disks leads to data loss. One string failure can leave data unavailable if at least three entries from a single reliability group reside there. The small number of states in the Markov chain in Figure 5 hints at the loss of resilence. The numerical values within the Figure reflect our twelve string model with 10 disks per string; many of generic expressions are too lengthy to display. If a disk failure occurs, a concurrent string failure almost always results in data loss. Consequentially the scheme is attractive if we can super-harden the strings by doubling the cooling. This relatively inexpensive step yield data survival rates comparable with the MDS array scheme 4D.

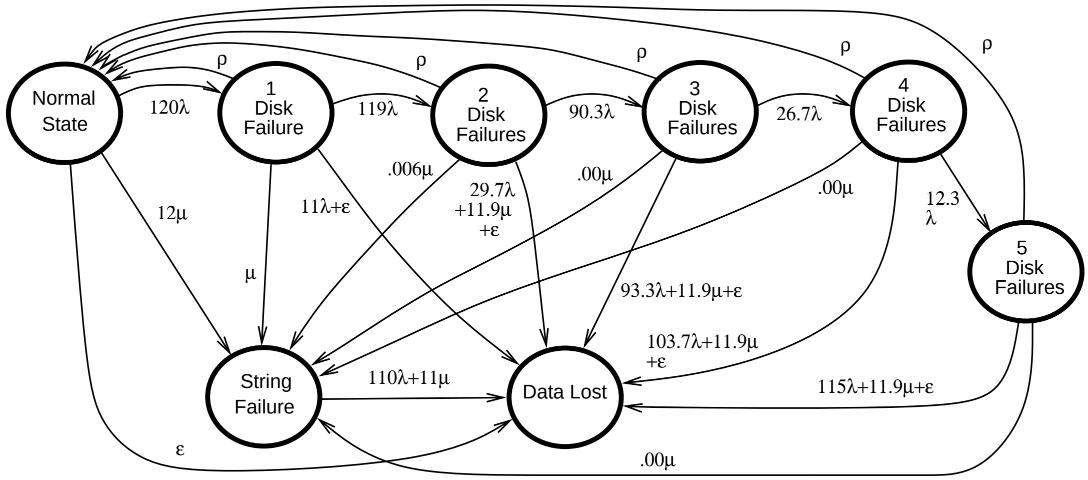

Figure 5: Markov Model for Dynamic Two Dimensional Parity Disk Array

For our third mode, we can use the two stage addressing scheme 4A to obtain better MTTDL figures. The addressing scheme, designated 5C, maps by 1) selecting a string and 2) then a disk within the string. This approach has slightly better resilience than 5B since we can ensure that each disk within a reliability group will reside on a different string thereby avoiding data loss from a single string failure.

We introduce 4 spares on their own string into the organization containing hardened strings. Our addressing schemes 5D and 5E uses the single step addressing of 4B and the double step addressing mode of 4C respectively. The major advantage of the two-dimensional parity scheme in these later incarnations over the MDS array is the ease of encoding the check data. However, the additional data redundancy compared to the generous use of spares in level 5 reliability groups would lead to greater resilience against semi-catastrophic failure and operational errors; the MDS scheme does retain some attractive aspects. These two schemes yield attractive MTTDL results as presented in Table 2.

As another alternative to super-hardened strings, we can improve the string layout to provide fault tolerance for even a pair of concurrent disk failures. This represents on going work; we present in Figure 6 a two dimensional array that will always allow reconstuction of the data in the presence of a pair of string failures. We have preliminary but very encouraging results regarding the MTTDL for this approach at the time of writing this abstract. The data organization problem remains open as we have organizations for a few small disk arrays. The MTTDL values need to be obtained as well.

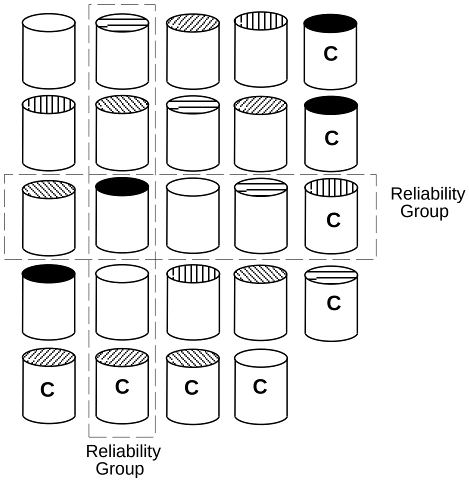

Figure 6: Two String Failure Resilient Parity Disk Array

6 Conclusions

We have investigated the reliability of several RAID organizations – the MDS array, the two-dimensional parity RAID and the Level 5 RAID with a string of spares all provide highly attractive

MTTDL results. Since the need for large ensembles of disk arrays together [1] with the attractive prices for disks, obtaining a 10,000 year MTTDL is not impossible. Nevertheless approximately 8% of the disk arrays will experience triple failure during a MTTDL interval [7]; we have not totally solved the problem. However our work represents a step forward.

To achieve these values, we can (1) provide spare strings, (2) shorten repair intervals, (3) harden disk drives, (4) harden strings or (5) increase data redundancy. We summarize the benefits and drawbacks with each.

The advantage of spares are the absence of performance penalties, which by distributing [5] can be turned actually into an advantage. The disadvantage is that reaction to failure has to be swift – we assumed instanteneous replacement of failed disks by spares – and that no protection against related, semi-catastrophic failures and against operation failures is provided.

MTTDL results are very sensitive to the repair rate. However the quick human response to failure comes at a price. The repair person with a disk drive in hand etc. waits to replace a spare and draws a salary throughout the interval.

Disk MTTF has increased considerably over the last years without increased costs to the consumer; It is difficult to predict whether this development will continue. However a disk contains mechanical parts and cannot be hardened arbitrarily.

Strings however can be provided with almost arbitrary degrees of redundancy. The component costs make this a reasonable first step for secure data storage.

Increasing data redundancy implies increasing the number of writes for a single storage operation. The use of a non-volatile cache keeps this performance degradation in more than tolerable bounds and is necessary for operation safety as well. Worst case performance is markedly increased compared with spares, as the adjustment to failed components is either unnecessary or can be done more leisurely to increase, for example, the read performance by swapping check data for original data. The two choices for this redundancy increase are the use of MDS codes or the two-dimensional parity scheme. The first choice uses a more complicated encoding scheme, which however has been used for safe data transfer as an error correcting code. The second choice is more vulnerable to string failures but can be designed using level 5 components. If strings are super-hardened, the MTTDL numbers are almost indistinguishable.

We also considered, in the last paragraph of section 5, a totally new approach to providing reliability using software rather than hardware. The basic combinatorial problem here very interesting and we are continuing our study. Preliminary results, not reported here, suggest this approach is very worthy.

References

[1] Walter A. Burkhard and Jai Menon. Disk Array Storage System Reliability. In *Digest of Papers The Twenty-Third International Symposium on Fault-Tolerant Computing*, pages 432–441, Toulouse, June 1993.

[2] Robert Geist and Kishor Trivedi. An Analytic Treatment of the Reliability and Performance of Mirrored Disk Subsystems. In *Digest of Papers The Twenty-Third International Symposium on Fault-Tolerant Computing*, pages 442–450, Toulouse, June 1993.

[3] Garth A. Gibson, Lisa Hellerstein, Richard M. Karp, Randy H. Katz, and David A. Patterson. Failure Correction Techniques for Large Disk Arrays. In *Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS III)*, pages 123–132, Boston, April 1989.

[4] Garth Alan Gibson. *Redundant Disk Arrays: Reliable, Parallel Secondary Storage.* PhD thesis, University of California, Berkeley, December 1990. Report UCB/CSD 91/613.

[5] Jai Menon and Richard L. Mattson. Distributed Sparing in Disk Arrays. In *Proceedings of the COMPCOM Conference*, pages 410–416, San Francisco, February 1992.

[6] David A. Patterson, Garth A. Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings SIGMOD International Conference on Data Management*, pages 109–116, Chicago, 1988.

[7] Martin Schulze, Garth A. Gibson, Randy H. Katz, and David A. Patterson. How Reliable is a RAID? In *Proceedings of the COMPCON Conference*, pages 118–123, San Francisco, 1989.

[8] Thomas J.E. Schwarz and Walter A. Burkhard. RAID Organization and Performance. In *Proceedings of the 12th International Conference on Distributed Computing Systems*, pages 318–325, Yokohama, June 1992.

[9] Tom Schwarz and Walt Burkhard. Multi-Dimensional Disk Array Reliability. Technical report, University of California, San Diego, October 1993. CS93-324.