Disk Array Storage System Reliability

Walter A. Burkhard

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

Jai Menon

Storage Architectures
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Abstract

Fault tolerance requirements for near term disk array storage systems are analyzed. The excellent reliability provided by RAID Level 5 data organization is seen to be insufficient for these systems. We consider various alternatives – improved MTBF and MTTR times as well as smaller reliability groups and increased numbers of check disks per group – to obtain the necessary improved reliability. The paper begins by introducing two data organization schemes based on maximum distance separable error correcting codes. Several figures of merit are calculated using a standard Markov failure and repair model for these organizations. Based on these results, the multiple check disk approach to improved reliability is an excellent option.

1 Introduction

Disk array storage systems, especially those with redundant array of independent disks (RAID) Level 5 data organization [13], provide excellent cost, run-time performance as well as reliability and will meet the needs of computing systems for the immediate future. We project the needs of computing systems to the year 2000 and beyond and find that even greater reliability than provided by RAID Level 5 may be needed.

Goldstein has collected data from moderate to large commercial installations and he projects that the average installation in 1995 will contain two terabytes (TB) of data [9]. We use his projection techniques to obtain results for the year 2000 and conclude the average commercial installation will have ten TB of data. We expect this projection is pessimistic since it does not account for the proliferation of data stored for video images of multimedia applications.

Currently the typical disk form factor is 5.25 inches but this is rapidly diminishing to 3.5 inches; already 20 MB 1.3 inch form factor disks are available. By year 2000, we expect the capacity of such disks to expand to at least 200 MB due to areal density increases. These small disks will be the typical storage media within commercial installations. Accordingly the average installation will have 50,000 drives. If we assume highly reliable disk drives with mean-time-between-failures (MTBF) of 500,000 hours and with mean-time-to-repair (MTTR) of one hour, the mean-time-to-data-loss (MTTDL) within a (10+P) RAID Level 5 array is 2.3×10^9 hours. Then the average ten TB installation, constituted of 5,000 such RAID Level 5 arrays, has a 460,000 hour MTTDL. That is, the installation MTTDL is just slightly less than its constitutient disk MTBF.

Consider 1000 such installations; the MTTDL of the group of installations is approximately 460 hours. That is every 460 hours, less than 20 days, some installation within this 1000 will lose data. As the number of drives per installation becomes large and as the number of such installations grows, even the excellent reliabilities provided by RAID Level 5 disk organizations will not suffice. While the RAID Level 5 data organization is sufficient for now, we need to consider schemes that provide higher reliabilities.

Our goal within this paper is to present and analyze a data organization scheme for disk arrays that provides better reliabilities than RAID Level 5. The data organization, derived from maximal distance separable (MDS) codes, retains some of the advantages of RAID Level 5 while providing higher reliability. The scheme generalizes the RAID data organization and is the most efficient data organization in the sense that for a given number of data disks it can maintain data integrity through c concurrent disk failures with exactly c check disks. Several schemes for providing better reliability than RAID Level 5 have been proposed. Gibson et al. present the "multidimensional" parity schemes [8]. Blaum et al. present a scheme that accommodates two concurrent disk failures [2] that is

also based on a novel variety of MDS codes [3]. Cheung and Kumar study "multi-parity" schemes that accommodate a fixed number of concurrent failures [6]. Neither the multidimensional or multi-parity schemes are MDS organizations; both require considerably more than c check disks to withstand c concurrent disk failures.

Our paper is organized of follows. Section 2 contains our presentation of two MDS data organization schemes. Within section 3 we begin with an approximation for the reliability functions for MDS schemes with one, two or three check disks. Then we present a derivation of the mean-time-to-data-loss for the MDS schemes. Next we present the data loss probability as a function of the number of installed disk arrays. We refine this result by determining the number of disk arrays that remain operational from the start until time t. In section 4 we explore several approaches to improving reliability such as increased MTBF or decreased MTTR. Our results indicate that the most effective way to improve the fault-tolerant capabilities of disk arrays is to include additional check disks. We conclude by observing that if we have several thousand disk arrays operational, we will want additional fault-tolerance provided by the two check disk MDS organizations.

2 MDS Disk Organization

The MDS disk organization is derived from the so-called maximal distance separable (MDS) codes that have been known for years in coding theory [11]. Independently similar schemes have been reported by Karnin, Greene, and Hellman[10] who study the sharing of information within a "faulty environment." Rabin[15] also reports an identical scheme in the context of message transmission. Abdel-Ghaffar and El Abbadi report a file comparison scheme that uses this approach [1]. Preparata [14] was one of the first to observe the connection between Rabin's approach and the MDS codes. Recently Blaum *et al.* [2] have created a novel MDS array coding scheme which we present as well.

MDS disk organizations are described in terms of two parameters n and m. MDS organizes a file into n equal-sized pieces, referred to as *fragments,* with each stored as an ordinary file on a different disk. The second MDS parameter m specifies the number of fragments such that any m suffice to reconstruct the file. The fragments are approximately $1/m^{\text{th}}$ the size of the file. We can construct the file if we have access to all n fragments, but this approach need not provide any fault tolerance. We could store, for example, every n^{th} file character within the same fragment and require all n fragments to construct the file. However if the fragments are each identical to the file itself, we have (re-named) file replication; this of course will provide excellent fault tolerance but with extravagant disk storage space requirements. The process of creating the n fragments is referred to as "file dispersal" and the process of recreating the contents of the file from m fragments is referred to as "file reconstruction."

The dispersal operation maps a fixed number of data character values to n fragment character values. We present two MDS organizations, one mapping m data byte values and the other mapping $m \times (n-1)$ data byte values. The first uses a dispersal matrix which is similar to the generating matrix from coding theory and the second uses horizontal and diagonal syndromes also from coding theory. It is an interesting task to verify that each scheme is based on MDS codes but we will not pursue this here. As examples, we present an $n = 5$, $m = 3$ configuration using each scheme; the size of each fragment will be approximately one-third the size of the file.

The first MDS organization utilizes an $m \times n$ dispersal matrix D to specify the file dispersal operation. The dispersal operation maps m data byte values to n fragment byte values. The dispersal matrix must have the property that any m columns are linearly independent; this condition assures the MDS property will hold and we refer to it as the *independence* property.

An example dispersal matrix is

$$D \quad = \quad \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 & 4 \end{pmatrix}.$$

There are many such dispersal matrices even for the $n = 5$, $m = 3$ configuration. File dispersal is accomplished using D such that three file characters are mapped to five fragment characters.

$$(p_{3j}, p_{3j+1}, p_{3j+2})\, D \quad = \quad (f_{0,j}, f_{1,j}, f_{2,j}, f_{3,j}, f_{4,j})$$

Fragment character $f_{i,j}$ is stored as the j^{th} character within fragment F_i. Continuing our example, suppose that our file contains the ASCII encoded text:

```
The old man and the sea.\n
```

Then the five fragments contain the following byte values expressed in octal notation: Fragments F_0, F_1 and F_2 consist of every third value in the file as demonstrated in figure 1. Fragments F_3 and F_4 store the sum and products of byte values as byte values without

F_0 :	124	040	144	141	141	040	145	145	012
F_1 :	150	157	040	156	156	164	040	141	000
F_2 :	145	154	155	040	144	150	163	056	000
F_3 :	131	043	051	057	153	074	066	052	012
F_4 :	141	077	341	075	134	031	230	037	012

F_0 :	T	\40	d	a	a	\40	e	e	\n
F_1 :	h	o	\40	n	n	t	\40	a	\0
F_2 :	e	l	m	\40	d	h	s	.	\0

Figure 1: Fragments F_0, F_1, and F_2

overflow using arithmetic within finite field $GF(2^8)$. This finite field is selected since a byte can assume one of $256 = 2^8$ values. Addition and subtraction operations are the exclusive-or operation; the multiplication and division operations could use tables created as described within [11].

The reconstruction process is defined as a transformation that maps byte values from m fragments to m data byte values for file. Since any m columns within D are linearly independent, we construct the $m \times m$ inverse matrix R for the columns associated with the m fragments participating within the reconstruction. The fragment entries are processed sequentially.

There are two varieties of fragments within our example; the three fragments F_0, F_1 and F_2 allow very easy file reconstruction since merging them is all that is necessary. Any other combination of fragments will require "real" computation during reconstruction. While the presence of fragments allowing this simplistic reconstruction is not required by the MDS data organization scheme, it is certainly advantageous. Dispersal schemes containing these ease-of-reconstruction fragments are said to have the *systematic* property. More generally, we will require that our dispersal matrices always have this property; that is, the dispersal matrix contains m columns constituting an identity matrix. Selecting a dispersal matrix satisfying both the independence and systematic properties is relatively straightforward as shown in [16]. Reconstruction will be the very straightforward merging of files for the m fragments associated with these columns. We refer to these m fragments as *primary fragments* and the others as *secondary fragments.* In the absense of failed sites and other conditions being equal (such as site load) the reconstruction process will access only the primary fragments.

We have two requirements for the dispersal matrices: (1) the systematic property and (2) the independence property. The independence property ensures that any m fragments suffice to reconstruct the file from the fragments. The systematic property ensures that for one combination of m fragments, reconstruction is accomplished by merging the fragments. The Vandermonde matrix [12] provides an $n \times n$ matrix having non-zero determinant thereby assuring linear independence of its columns. We can construct dispersal matrices by truncating the bottom $n - m$ rows of the Vandermonde matrix. Since any m shortened columns constitute an $m \times m$ Vandermonde matrix, any combination of m columns will be linearly independent as required. We then utilize elementary matrix transformations to obtain a dispersal matrix in which the leftmost m columns form the identity matrix. We can construct such a matrix over $GF(2^8)$ with as many as 256 columns. Going full circle, we utilize results from coding theory regarding twice-extended Reed Solomon codes [11] to obtain suitable dispersal matrices over $GF(2^8)$ with 257 columns. For practical applications this is probably more than adequate. We note that our previous dispersal matrix was not selected in this fashion.

The Blaum *et al.* MDS scheme is based on array codes; we present an $n = 5$, $m = 3$ configuration example here as well. Dispersal maps 12 data byte values to 20 fragment values. Codewords are 4×5 arrays

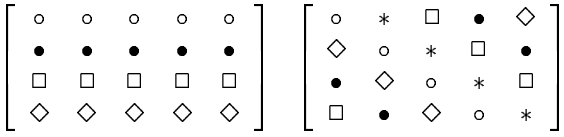

Figure 2: Array Code Parity Matrix Schema

in which each entry is a byte value. The byte values within each column are stored on a disk drive. The parity conditions (syndromes) are listed here for codeword $C = \{c_{i,j}\}$ where $0 \le i \le 3$ and $0 \le j \le 4$:

$$\bigoplus_{j=0}^{4} c_{i,j} = 0$$

$$\bigoplus_{k=0}^{3} c_{k,(j+k) \bmod 5} = 0$$

The first equation defines the horizontal syndromes and the second the diagonal syndromes. The pair of arrays within figure 2 geometrically define the parity

constraints for our code. In our example, each codeword designates 20 byte values stored four per disk on five disks. Twelve data character values are positioned within the leftmost three columns of the matrix; this is the systematic property for these codes. The rightmost two columns contain the parity information. We continue our example and suppose that our file contains the ASCII encoded text:

```
The old man and the sea.\n
```

Then the five fragments contain the following byte values expressed in octal notation:

```
124 040 144 141 141 040 145 145 012 000 000 000 : F0
150 157 040 156 156 164 040 141 000 000 000 000 : F1
145 154 155 040 144 150 163 056 000 000 000 000 : F2
171 120 175 126 105 175 112 146 012 012 012 012 : F3
040 163 124 171 056 101 174 114 000 012 012 012 : F4
```

Fragments F_0, F_1 and F_2 consist of every third value in the file as demonstrated in figure 3. Fragments F_3 and F_4 contain the parity values according to the eight equations presented above.

```
T \40  d  a  a \40  e  e \n \0 \0 \0 : F0
h  o \40  n  n  t \40  a \0 \0 \0 \0 : F1
e  l  m \40  d  h  s  . \0 \0 \0 \0 : F2
```

Figure 3: Fragments F_0, F_1, and F_2

Reconstruction of the data should one or two disks fail proceeds in a two phase process. First we calculate the horizontal and diagonal syndromes assuming two disks have failed; the associated columns within the codewords contain zero entries. The second phase determines the values in the two "erased" columns.

This scheme utilizes only the exclusive-or operation and it is applicable for disk arrays containing a prime number of disks; it is discussed with far greater generality within [2, 3]. One generalization is the number of concurrent faulty disks accommodated and another relaxes the restriction to only prime numbers of disks per array.

3 Failure Models

Gibson presents a characterization of disk failures [7] from which he concludes that an exponential lifetime distribution is a reasonable approximation especially for disk units with mean time to failure of at least 200,000 hours. He compares the exponential and Weibull distributions for disk units having smaller expected lifetime figures. In this situation, he concludes that neither distribution adequately models these lifetimes and possibly a three parameter distribution is necessary. However Gibson states that the exponential distribution is plausible throughout.

The infant mortality phenomena within disk units has been observed. We have anecdotal evidence stating that one-half of the failures, occurring within a nominal lifetime span for disk units, occur within the the first four months of operation. We have modeled this effect by piecing together two exponential lifetime distributions; the difference between using a single and a pair of exponential distributions is very small within our results. Accordingly we present results for a single exponential distribution lifetime model.

The Markov model depicted in figure 4 is a simple model sufficient for reliability groups containing $n = m + c$ disks and accommodating up to c concurrent disk failures without data loss. We assume disk

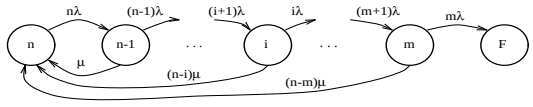

Figure 4: MDS Markov Model

failures are independent and that both the MTBF and MTTR for the disks is time independent. It is convenient to use the reciprocals of these quantities, which are rates of change rather than time intervals, within our model – the failure rate per disk λ is MTBF^{-1} and the repair rate per disk μ is MTTR^{-1}. The state labels designate the number of operational and accessible fragments. Associated with each state is probability $P_i(t)$ designating the probability of being in state i at time t. The failure state F is absorbing in our model since it is entered only if fewer than m fragments are accessible. Other Markov models are possible here. We could repair a single disk at a time, but the repair process produces all fragments at once. To a first approximation, this variety of model change only slightly affects the reliability values and not our conclusions.

In this section, we present three measures of fault tolerance suitable for disk arrays. We begin by presenting an approximation to the reliability function derived from the Markov model within figure 4. We utilize this approximation in section 4. Our first fault tolerance measure is the traditional mean-time-to-data-loss (MTTDL). Both the second measure, the

data loss probabilty (DLP), and the third measure, the expected number of failures, refine the MTTDL measure. In all cases, improving the reliability improves our result. The second and third measures are based on ensembles of disk arrays.

3.1 Mean Time To Data Loss

The mean time to data loss MTTDL is the expected time to enter state F. A similar derivation obtaining approximate values is presented in the USENIX Proceedings [5]. The reliability function $R_{c,m}(t)$ is the probability of being in one of the non-failure states at time t; that is

$$R_{c,m} = P_{m+c}(t) + \cdots + P_{m+1}(t) + P_m(t). \quad (1)$$

We adopt the notation $R_{c,m}(t)$ to emphasize the uniformity assumed of the disk arrays, each array contains m data and c check disks; the state probability notation requires a second subscript as well but we choose to rely on the context for these notations. The mean time to data loss for the configuration with c check and m data disks is

$$\text{MTTDL}_{c,m} = \int_0^\infty R_{c,m}(t)\,dt. \quad (2)$$

We determine $\text{MTTDL}_{c,m}$ using the Laplace transformation "integral property." Then

$$\text{MTTDL}_{c,m} = R^*_{c,m}(0) = \sum_{i=m}^{m+c} P^*_i(0)$$

where R^* and P^*_i designate transformed functions. We present a specific case here and the general result in [4]. The Laplace state transition equations for $c = 2$ are

$$\begin{aligned} P^*_{m+2}(s) &= (s + \lambda(m+1) + \mu)(s + \lambda m + 2\mu)/\Delta(s) \\ P^*_{m+1}(s) &= \lambda(m+2)(s + \lambda m + 2\mu)/\Delta(s) \\ P^*_m(s) &= \lambda^2(m+1)(m+2)/\Delta(s) \end{aligned}$$

where $\Delta(s)$ is

$$(s + \lambda(m+2))(s + \lambda(m+1) + \mu)(s + \lambda m + 2\mu) - 2\mu\lambda^2(m+1)(m+2) - \mu\lambda(m+2)(s + \lambda m + 2\mu).$$

Finally we obtain the following

$$\text{MTTDL}_{2,m} = \frac{2\mu^2 + \mu\lambda(5m+6) + \lambda^2(3m^2+6m+2)}{\lambda^3 m(m+1)(m+2)}.$$

For reliability modeling such as the above, the ratio μ/λ is very large; accordingly we can approximate the mean time to data loss expressions by the following [5] for $c = 0, 1, 2, \cdots$

$$\text{MTTDL}_{c,m} = \frac{\omega^c}{\lambda m \binom{m+c}{c}} \quad (3)$$

where $\omega = \mu/\lambda$. The improvement in $\text{MTTDL}_{c,m}$ as a function of c is rather dramatic. The mean time to data loss values are given in Table 1 where

c	MTTDL years
0	2.8
1	2.6×10^5
2	4.3×10^{10}
3	1.0×10^{16}

Table 1: Mean Time to Data Loss Results

$m = 10$ data disks per array, $\lambda = 4 \times 10^{-6}$ failures per hour, and $\mu = 4$ repairs per hour. These results are very impressive; incrementing c increases the expected longevity by approximately ω which is 10^6 in this example. However as we will observe in the next sections, these figures provide too rosy a view on the performance of an ensemble of disk arrays.

3.2 Reliability

Our measures of disk array fault tolerance depend upon the reliability $R_{m,c}(t)$ of the array. The process of determining these values is tedious at best and we seek an accurate approximation. For $c = 0$, the reliability is

$$R_{0,m}(t) = e^{-t\lambda m} = e^{-t/MTTDL_0} \quad (4)$$

In the cases $c > 0$, we have Theorem 1 which contains approximations to the reliability function; we derive and verify this approximation in [4].

Theorem 1 *For each disk array containing m data disks and $c = 1, 2,$ or 3 check disks with failure rate λ and repair rate μ we have the following approximate reliability functions*

$$R_{1,m}(t) = C_{1,m,0}\, e^{-t/\text{MTTDL}_{1,m}} + C_{1,m,-\mu}\, e^{-t(\mu+\lambda(2m+1))} \quad (5)$$

$$R_{2,m}(t) = C_{2,m,0}\, e^{-t/\text{MTTDL}_{2,m}} + C_{2,m,-\mu}\, e^{-t(\mu+\lambda(2m+3))} + C_{2,m,-2\mu}\, e^{-t(2\mu+\lambda m)} \quad (6)$$

$$R_{3,m}(t) \quad (7)$$
$$= C_{3,m,0}\, e^{-t/\mathrm{MTTDL}_{3,m}} + C_{3,m,-\mu}\, e^{-t(\mu+\lambda(2m+5))} + C_{3,m,-2\mu}\, e^{-t(2\mu+\lambda(m+1))} + C_{3,m,-3\mu}\, e^{-t(3\mu+\lambda m)}$$

where $MTTDL_{1,m}$, $MTTDL_{2,m}$ *and* $MTTDL_{3,m}$ *are given in equation 3 for* $c = 1, 2,$ *and* 3,

$$C_{c,m,0} = 1 + H_c m \left(\begin{array}{c} m+c \\ c \end{array} \right) \left(\frac{\lambda}{\mu}\right)^{c+1}$$

and H_c *is the* c^{th} *harmonic number. The other coefficients are*

$$C_{1,m,-\mu} = -\frac{\lambda^2 m(m+1)}{\mu^2},$$
$$C_{2,m,-\mu} = -\frac{\lambda^3 m(m+1)(m+2)}{\mu^3},$$
$$C_{2,m,-2\mu} = \frac{\lambda^3 m(m+1)(m+2)}{4\mu^3},$$
$$C_{3,m,-\mu} = -\frac{\lambda^4 m(m+1)(m+2)(m+3)}{2\mu^4},$$
$$C_{3,m,-2\mu} = \frac{\lambda^4 m(m+1)(m+2)(m+3)}{4\mu^4},$$
$$C_{3,m,-3\mu} = -\frac{\lambda^4 m(m+1)(m+2)(m+3)}{18\mu^4}.$$

This approximation is valid provided λ/μ is very small. The coefficients are determined using the partial-fraction expansion of the transformed reliability function. The roots of the denominator are approximated using Newton's iteration scheme. Moreover, the largest discarded term in each coefficient is smaller, in magnitude, by a factor of λ/μ.

3.3 Data Loss vs Number of Units

Our second figure of merit is obtained by considering a large ensemble of disk arrays. We desire each array not lose data during the typical lifetime of the constituent disk arrays, since loss of any data, however small, is highly traumatic for the users losing data. The data loss probability $DLP_{c,m}(n,t)$ depends on the number of disk arrays n within the ensemble, the number of check disks per array c, the number of data disks per array m as well as the time interval t. We consider only uniform ensembles of arrays in this analysis; moreover we assume that array failures are independent. Then we have

$$DLP_{c,m}(n,t) = 1 - R_{c,m}(t)^n \quad (8)$$

measuring the probability of at least one data loss within the interval from the start to time t. We plot $DLP_{c,m}(n,t)$ for fixed t in figures 5, 6, 7 and 8. The three sets of curves in figure 5 have the same parameters except for c; each array constains 10 data disks and the time interval is 7 years. The individual disk failure rates $\lambda = 1/100000, 1/200000, 1/300000, 1/400000$ and $1/500000$ failures per hour. The five curves, in the upper left-hand corner of the graph, immediately to the left of the phrase c = 0 designate no check disk within the arrays. The leftmost curve has probability of data loss greater than 0.9978 for one array; this curve appears almost as a horizontal line within Figure 5. The probability of failure is practically one for as few as ten disk arrays for any of our five failure rates when c is zero. The five curves immediately to the left of the phrase c = 1 designate single check disk probabilities and the curves to the right of c = 2 pertain to two check disks. Within each set, the curves from left to right have decreasing λ values. Thus for example, with $n = 10000$ arrays each configured with a single check disk, we expect at least one data loss with probability approximately 2/5 for $\lambda = 1/200000$. However utilizing two check disks per array, the probability of data loss is very close to zero for the same number of arrays.

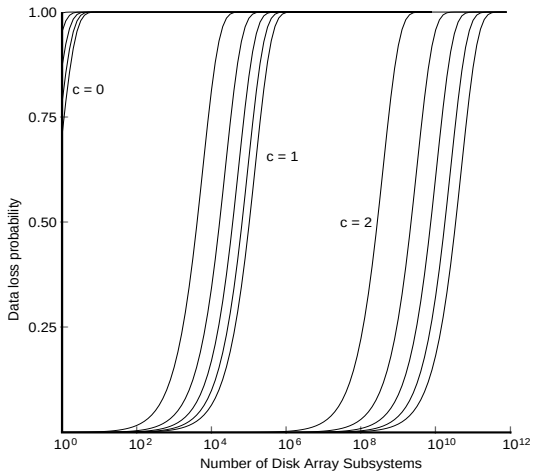

Figure 5: Data Loss vs Number of Subsystems

3.4 Expected Number of Failures

The probability of data loss depends on whether our array configurations contain two, one or no check disks. We are interested in reducing the expected number of "unhappy" clients who lose data during the disk array lifetime. We propose to refine our measure by determining the expected number of ar-

rays that will lose data during the observation interval. Let $X_{c,m}(t,n)$ designate the number of arrays, each containing c check disks and m data disks, remaining operational at time t provided that n are initially placed in service. Then $prob\{X_{c,m}(t,n) = i\}$ for $i = 0, 1, \ldots, n$ is

$$\binom{n}{i} R_{c,m}(t)^i (1 - R_{c,m}(t))^{n-i}.$$

Since $X_{c,m}$ has a binomial distribution, the expected number of array remaining operational at time t is $nR_{c,m}(t)$. Thus we can interpret $R_{c,m}(t)$ as the expected fraction of the arrays remaining operational during the interval from the start until time t. The expected number of failures over a seven year period for an ensemble of 10,000 disk arrays each with 10 data disks, with a failure rate per disk of 5×10^{-6} per hour, and repair rate of 4 disks per hour are tabulated within Table 2. The expected number of failures becomes insignificant only when c is at least two for repair and failure rates typical of current technologies. We conclude that RAID Level 5 ($c = 1$) may not be a satisfactory storage architecture for massively employed subsystems since the probability of at least one unhappy client is not small.

c	Expected Failures
0	9535
1	0.42
2	5×10^{-6}

Table 2: Expected Failures

4 Parameter Sensitivity

If a given configuration with one check disk is not reliable enough, we desire to improve it. Using additional check disks (increasing c) is only one of several alternatives to improving its reliability. Other approaches include manufacturing disk drives with greater MTBF (decreasing λ), repairing failed disks faster with smaller MTTR (increasing μ) and using smaller reliability groups (decreasing m). So we need to explore the effects of incremental changes in λ, μ, m or c individually have on the reliability. We can calculate partial derivatives of $R_{c,m}$ with respect to λ and μ as well as differences with respect to m and c. We are left with the question of trade-offs among the reliability improvements and the costs of the incremental changes. We present curves demonstrating the sensitivity of the parameters and then conclude with some analytical results. Within figure 5 we have already noted the significant improvement in the data loss probabilities when c is incremented.

We begin with one check disk configurations. Figure 6 demonstrates the effect of various failure rates with other parameters constant. The failure rates, failures per hour, are written diagonally across the curves; each disk array contains ten data disks and one check disk, the repair rate is four disks per hour and the observation interval is seven years. A typical failure rate, within current technology, is 1/500000 per hour; this curve rapidly goes to one at approximately 10^4 disk arrays. We next consider various repair rates, very slow to "impossibly" fast, and present the results within figure 7. The repair rates, disk repairs per hour, are written diagonally across the curves. Repair rates much larger than four per hour are not currently possible. Each disk array contains ten data and one check disk and the observation interval is seven years. The incremental improvement for the data loss probability is less for repair increments than for failure decrements.

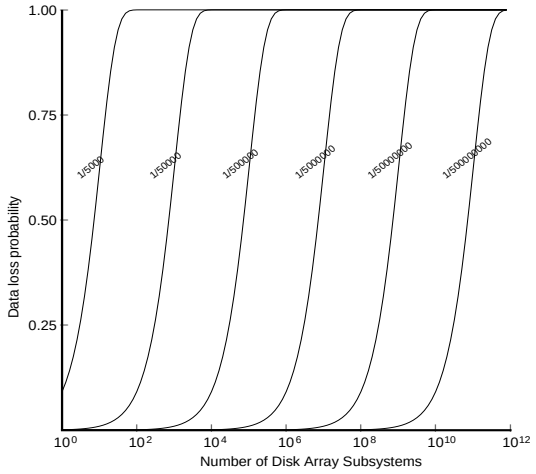

Figure 6: Probability of Data Loss vs Number of Subsystems: Various Failure Rates

We conclude our presentation of one check configurations with the curves of figure 8 in which we explore the effects of varying m on the reliability. We maintain constant data storage capability and each subsystem contains exactly 120 data disks. The data disks are logically partitioned into disjoint arrays. Each component array contains a single check disk, its repair

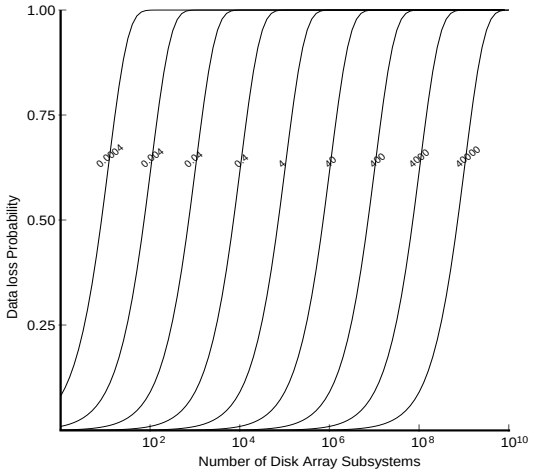

Figure 7: Probability of Data Loss vs Number of Subsystems: Various Repair Rates

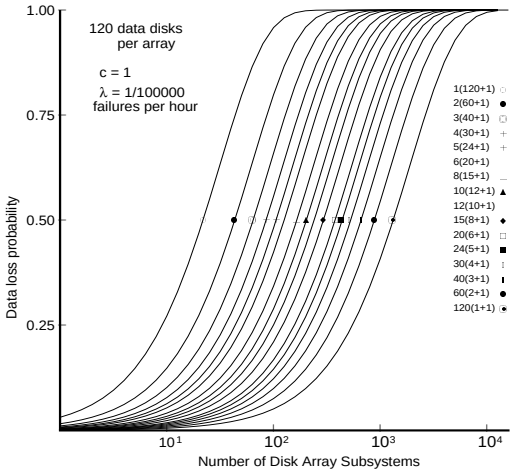

Figure 8: Probability of Data Loss vs Number of Subsystems: Constant Data Disk Capacity

rate is four per hour and the observation interval is seven years. For example, we might have one array with 120 data disks and one check disk or we could have 2 arrays each containing 60 data disks and one check disk and so forth. The notation i(m+1) designates i arrays each with m data disks; the product m × i is always 120. The 120(1+1) configuration designates data replication, there are 120 check disks within this configuration. The probabilities of data loss specified within figure 8 are for organizations each containing 120 data disks comprised of a number of disk arrays. The probability of data loss diminishes with increased numbers of check disks within the ensemble.

Next we consider two check disk configurations. We present figures 9, 10 and 11 which contain the data loss probabilities for various failure rates, repair rates, and the constant data capacity configurations respectively; each is for a seven year observation interval. We note within that the curves are spread further apart than for the single check disk curves. The extra check disk increases the sensitivity to changes in the failure rate; that is, small changes in failure yield larger data loss probability improvements within figure 9 than in figure 6. We observe a similar behavior with respect to the repair rates in figure 10. We observe again the improved sensitivity to increases in the repair rate; the curves are spread further apart than for the single check disk curves. Finally, we present constant storage capacity curves in figure 11. These curves spread over a very wide range providing improved sensitivity to increased numbers of check disks. The notation i(m+2) within the figure designates a group of i disk arrays each containing m data disks and two check disks such that i × m is 120.

Within figure 11, the repair rate is four disks per hour and the observation interval is seven years.

We consider the improvement obtained due to an extra check disk per disk array versus partitioning the disks into smaller groups. For example, we consider within figure 8 the 1(120+1) and 2(60+1) curves. The pair of smaller groups has a slightly lower probability of data loss values. However, if we consider the 1(120+2) configuration within figure 11 we note the probability of data loss dominates either of the previous curves. Adding two more disks to obtain a 2(60+2) configuration is even better. We note that *all* the one check disk curves are dominated by each two check disk curves. For one check disk, the probability of data loss is very close to one for 10^4 subsystems; for two check disks, the probability is very close to zero for otherwise identical parameters.

5 Conclusions

Many commercial applications are projected to require 10 terabytes of on-line storage by year 2000. Even with projected improved disk form factors and storage densities, the reliabilities provided by RAID Level 5 architectures are not sufficient for configurations with this number of disk drives. The storage sizing projections are very conservative, without including video image storage requirements.

We have presented the MDS disk array architec-

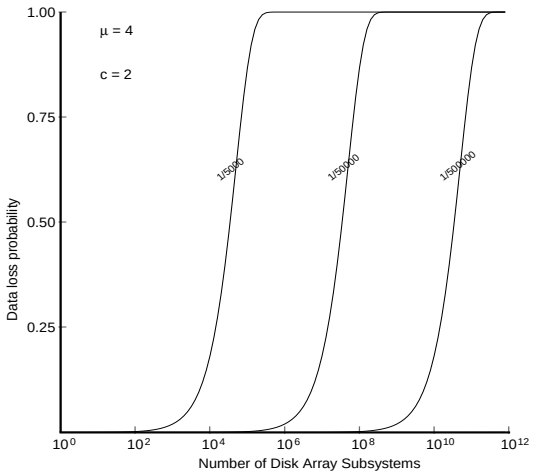

Figure 9: Probability of Data Loss vs Number of Subsystems: Various Failure Rates

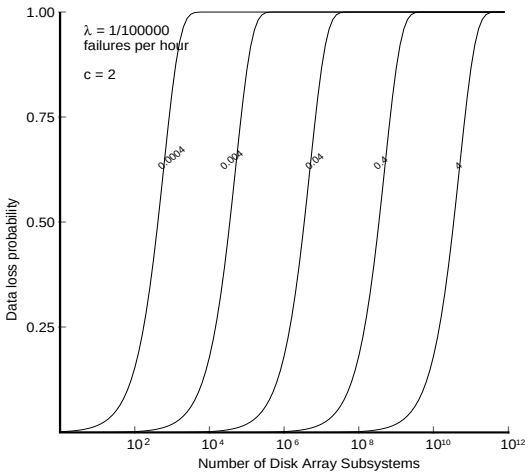

Figure 10: Probability of Data Loss vs Number of Subsystems: Various Repair Rates

tures capable of providing reliabilities arbitrarily close to one. These schemes utilize the smallest number of additional disks c to ensure data integrity in spite of up to c disk failures. Two MDS schemes are outlined – the first based on Reed-Solomom error correcting codes and the second on Blaum-Roth error correcting codes.

We considered various parameters in our reliability study; the failure and repair rates for disk drives and the number of check and data disks in a disk array. Figure 5 indicates the dramatic improvements to the data loss probability obtained by incrementing c. The probability of data loss depends exponentially on the reliability as specified in equation 12. Increasing the reliability lowers the data loss probability. The ensemble of one check disk data loss probability curves, for RAID Level 5, are approximately one for approximately 10^6 disk arrays. However for two check disks, the data loss probability is approximately zero for the same number of disk arrays. We conclude that by year 2000 the additional check disk is called for within large-sized storage systems.

The figure 5 curves also provide insights regarding the sensitivity of the data loss probability to changes in the failure rate λ. Figures 6 and 9 elaborate this study; in figure 6 the failure rates are varied from 2×10^{-4} to 2×10^{-9}. The range 2×10^{-4} to 2×10^{-6}, in figure 9, provides equally good probabilities of data loss. The 2×10^{-6} rate is typical of current technology; within the $c = 2$ configuration, the probabilities of data loss is approximately zero for up to 10^8 disk arrays while for $c = 1$ the failure rate must be diminished to 2×10^{-9} to obtain approximately the same result.

Figures 7 and 10 presents the results of varying the repair rate μ while holding the failure rate constant. Within figure 7, the repair rate varies from 4×10^{-4} to 4×10^4; rates above 4 are beyond current technologies. Within figure 10, the repair rate varies from 4×10^{-4} to 4. Considering the repair rate 4, within figure 7 one check disk configurations, the probability of data loss is approximately zero for less than approximately 10^3 disk arrays. Within figure 10 two check disk configurations, the probability of data loss remains approximately zero for less than approximately 10^6 disk arrays.

Figure 8 compares several constant storage configurations comprised of disk arrays each with one check disk. Figure 11 is similar except each array contains two check disks. All configurations contain 120 data disks; the number of disk arrays ranges over $1, 2, 3, 4, \ldots, 30, 40, 60, 120$. Within figure 8, for approximately 10^4 ensembles of disk arrays, the probability of data loss is approximately one. Within figure 11, for less than 10^3 and up to 10^4 for all curves except one, the probability of data loss is approximately zero. In these configurations, the repair rate μ is 4 and failure rate λ is 1×10^{-5} a rate typical of current technology.

We have considered four approaches to improving the reliability; however, we have not considered the relative costs of these actions. Decreasing the failure rate (increasing MTBF) represents an improvement in the manufacturing techniques for disks. Increasing the repair rate (decreased MTTR) requires improved

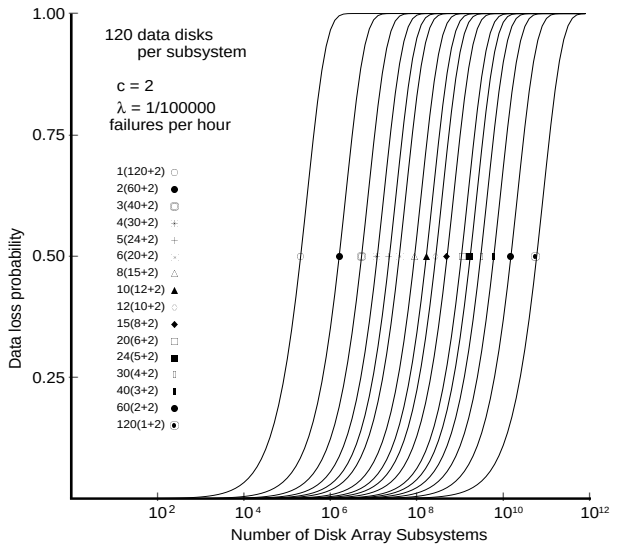

Figure 11: Probability of Data Loss vs Number of Subsystems: Constant Data Disk Capacity

disk seek, latency or transfer times. Changes in the number of data storage disks within a disk array have moderate effect on the probability of data loss; the total number of disks required increases as the size of the reliability groups decreases. We have not tried to optimize the reliability as a function of incremental costs. An optimization we have not considered in this study is the run-time performance of the disk arrays as a function of m and c. We expect that increasing c will have negative effect on run-time performance. We conclude that the two check disk configurations provide excellent fault tolerance. We need only address the run-time performance issue; this is the subject of our continuing research in disk arrays.

Acknowledgements

We wish to acknowledge conversations with Richard Mattson regarding the use of highly reliable disk arrays for applications requiring large numbers of storage systems as well as discussions with Thomas Schwarz and Mario Blaum regarding maximum distance separable error-correcting codes.

References

[1] Khaled A.S. Abedl-Ghaffar and Amr El Abbadi. An Optimal Strategy for Comparing File Copies. Technical report, University of California, Davis and Santa Barbara, 1992.

[2] Mario Blaum, Hsieh Hao, Richard L. Mattson, and Jai Menon. A Coding Technique for Recovery Against Double Disk Failures in Disk Arrays, 1989. US Patent Docket #SA889-0443 pending.

[3] Mario Blaum and Ronald M. Roth. New Array Codes for Multiple Phased Burst Correction. Technical Report RJ 8303 (75664), IBM Almaden Research Center, August 1991.

[4] Walt Burkhard and Jai Menon. MDS Disk Array Reliability. Technical Report CS92-269, University of California, San Diego, December 1992.

[5] Walter A. Burkhard and Petar D. Stojadinović. Storage-Efficient Reliable Files. In *Proceedings of the Winter 1992 USENIX Conference*, pages 69–77, San Francisco, January 1992.

[6] Shun Yan Cheung and Akhil Kumar. Multi-Parity: A Low Overhead Method for Achieving High Data Availability. Technical report, Cornell and Emory Universities, 1992.

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

[8] 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.

[9] Stephen Goldstein. Storage Performance – An Eight Year Outlook. In *Proceedings of the Computer Measurement Group*, pages 579–585, 1987.

[10] Ehud D. Karnin, Jonathan W. Greene, and Martin E. Hellman. On Secret Sharing Systems. *IEEE Transactions on Information Theory*, 29:35–41, 1983.

[11] Florence Jesse MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*. North Holland, 1978.

[12] L. Mirsky. *An Introduction to Linear Algebra*. Dover Publishers, New York, 1982.

[13] 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.

[14] Franco P. Preparata. Holographic Dispersal and Recovery of Information. *IEEE Transactions on Information Theory*, 35:1123–1124, 1989.

[15] Michael O. Rabin. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. *Journal of the Association for Computing Machinery*, 36:335–348, 1989.

[16] 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.