Designing Disk Arrays for High Data Reliability

Garth A. Gibson
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsbugh PA 15213

David A. Patterson
Computer Science Division
Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720

This research was funded by NSF grant MIP-8715235, NASA/DARPA grant NAG 2-591, a Computer Measurement Group fellowship, and an IBM predoctoral fellowship.

Proposed running head: Designing Disk Arrays for High Data Reliability

Please forward communication to:

Garth A. Gibson
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh PA 15213-3890
412-268-5890
FAX 412-681-5739
Garth.Gibson@cs.cmu.edu

ABSTRACT

Redundancy based on a parity encoding has been proposed for insuring that disk arrays provide highly reliable data. Parity-based redundancy will tolerate many independent and dependent disk failures (shared support hardware) without on-line spare disks and many more such failures with on-line spare disks. This paper explores the design of reliable, redundant disk arrays. In the context of a 70 disk strawman array, it presents and applies analytic and simulation models for the time until data is lost. It shows how to balance requirements for high data reliability against the overhead cost of redundant data, on-line spares, and on-site repair personnel in terms of an array's architecture, its component reliabilities, and its repair policies.

Recent advances in computing speeds can be matched by the I/O performance afforded by parallelism in striped disk arrays [12, 13, 24]. Arrays of small disks further utilize advances in the technology of the magnetic recording industry to provide cost-effective I/O systems based on disk striping [19]. But because arrays of small disks contain many more components than do larger single disks, failure rates can be expected to rise. For most users, increased failure rates are incompatible with secondary storage systems because secondary storage is thought to be the stable part of a computer system – the part expected to survive periodic malfunctions.

In our information society, malfunctions in computational components threaten the process by which new information is generated. But losses suffered by long-term storage components are even more debilitating because these destroy assets. It is no surprise then that institutions often consider the reliability of long-term storage crucial to their operation. Unfortunately, increasing reliability usually also increases cost.

The goal of this paper is to facilitate the cost-effective design of reliable secondary storage by developing and applying analytic models of the reliability of redundant disk arrays employing ‘‘N+1-parity’’ protection. This form of protection maintains the parity of N data disks in a single parity disk (possibly distributed evenly over all N+1 disks) that can be used to recover the contents of any single failed disk [19]. The models include a wide spectrum of disk array designs so that individual designers will be able to characterize the reliability of the system they want to build. We use Markov-model-solving software and simulation in this paper largely to validate the analytic models we present.

In this paper we present and apply four models for the reliability of redundant disk arrays that correct all single disk failures. The most fundamental model considers the effect of independent, random disk failures on an array’s data lifetime. The lifetime of data in an array ends when a failed disk’s data is lost. This first reliability model is based on a well-studied Markov model and yields a simple expression for reliability. A cost-effective method for improving reliability by maintaining a small number of on-line spare disks is addressed in a second, more complex model. It yields an analytic expression for reliability by solving separate submodels for data loss derived from spare-pool exhaustion and concurrent, independent disk-failures. A third model uses similar methods to address dependent disk failures induced by sharing interconnect, controller, cooling, and power-supply hardware (collectively called support hardware). Although N+1-parity protection only insures the correction of a single disk in a parity group, disk arrays can be organized so that each disk in a support-hardware group is contained in a distinct parity group. In this way, dependent disk failures are tolerable because they affect at most one disk per parity group. Finally, the fourth model bounds the reliability of disk arrays that incorporate on-line spare disks with dependent and independent disk failures. These bounds allow estimates of reliability with no on-line spares and with sufficient on-line spares to provide one- and two-spare, support-hardware groups. These four models show how disk arrays can provide high reliability with modest amounts of redundancy.

We use these models and simulations to explore the cost-reliability tradeoffs between arrays with different levels of redundancy. Traditionally, redundancy has been provided by full duplication, or mirroring, without on-line spares. Although this type of organization provides higher reliability than an N+1-parity organization of the same user capacity without spares, the addition of a few spares reverses this relationship. One of the most important results of this paper, specific to the design of practical disk arrays, is that an N+1-parity disk array with a few spares can yield greater reliability at lower cost than traditional mirrored disk arrays.

Throughout this paper differences in reliability models will be exemplified by their effects on an array of 70 3½-inch disks. Table 1 shows that this disk array is selected to match the capacity of an IBM 3390 disk subsystem with 70 IBM 0661 (Lightning) disks. Because the 3390 is IBM's newest, largest, and most expensive disk product, there is a lucrative market for a disk array that can exceed the performance and reliability of the IBM 3390 while matching its cost per megabyte.

Without redundancy, this example disk array unfortunately has virtually no chance of surviving three years without data loss because of the aggregate failure rate of its large number of components. With as little as 10% overhead for parity information, however, this disk array can be made about as reliable as a single disk. Then, if the failure of support hardware does not damage multiple disks simultaneously, the addition of a single on-line spare disk yields a mean time to loss of data that is about 10 times larger than a single disk. With two on-line spare disks, the mean time to loss of data in this disk array is marginally less than if it had an infinite number of on-line spare disks, specificly, about a factor of 20 times larger than a single disk. Even if this disk array is subject to dependent disk failures, an orthogonal arrangement of parity groups and support hardware groups and a single on-line spare disk yield about an 80% chance that data is not lost in 10 years. If this is not satisfactorily larger than the 56% chance that a single disk survives 10 years without data loss, raising the overhead for on-line spares to 10% allows failed support hardware to be replaced immediately and delivers a 98.7% chance of surviving 10 years without data loss.

A more thorough examination of these models and their impact on the design of disk arrays, as well as the lifetime distributions of magnetic disks and the performance and encoding of redundant disk arrays, appears in Gibson's dissertation [10].

1. Reliability Metric

The *Reliability*[1] of a system is defined for any target lifetime, t, as the probability that an individual system

[1] The use of the term *reliability* as a mathematically defined metric and as an intuitive concept can lead to ambiguity. We have chosen to follow the former convention and avoid the latter use.

survives for time t given that it is initially operational [27]:

$$R(t) = \text{Prob}(\, lifetime > t \mid initially\ fully\ operational \,)\ . \tag{1}$$

In stochastic terminology, $R(t) = 1 - F(t)$, where F is the cumulative distribution function (CDF) of system lifetimes:

$$F(t) = \text{Prob}(\, lifetime < t \mid lifetime > 0 \,) = 1 - R(t)\ . \tag{2}$$

In this work, *survival* means all user data is available or recoverable, so the reliability at time t is the proportion of systems that have not lost any user data in time t.

Because a function can be a cumbersome metric, reliability is frequently quoted as a simple probability with an implied time interval. For example, designers may be most interested in a system's one-year reliability ($R(1year)$), or, for the pessimistic among us, the probability that it will survive the duration of its warranty. Where we do not have a complete description of $R(t)$ or where its presentation involves too much data, we use estimates of the system's 1-, 3-, and 10-year reliabilities.

Perhaps the most commonly encountered measure of a product's reliability is its *Mean Time To Failure, MTTF,* or its *Mean Time Between Failures, MTBF.* Unfortunately, this metric does not give much information unless the lifetime distribution is known. Fortunately, our examination of 1350 5¼-inch disks in the field for up to 18 months shows little strong evidence against an exponential model for disk lifetimes [10], so the $MTTF$ of a disk drive may be assumed to imply a complete reliability function. Even without this assumption, our next section will explain why it is reasonable to expect that the lifetimes of practical disk arrays will have an approximately exponential distribution [1], so that the equivalent metric, *Mean Time To Data Loss, MTTDL,* is also a complete description of a disk array's reliability. Because the exponential lifetime distribution plays a prominent role in this paper, we include a close examination of its characteristics.

Where lifetimes are distributed exponentially as random variables, $R(t)$ has an exceptionally simple form fully determined by the product's mean lifetime, M:

$$R_{\text{exp}}(t) = e^{-t/M}\ . \tag{3}$$

Figure 1 shows the reliability of a system with exponential lifetimes as a function of time, which is expressed as a fraction of the system's mean lifetime. There is a 10% chance that this system will have a lifetime less than one-tenth its mean lifetime, a 37% chance of surviving one mean lifetime, and only a 5% chance of surviving three mean lifetimes.

Figure 2 shows the reliability over 1, 3, and 10 years for a system with exponential lifetimes as a function of its mean lifetime. This figure underscores the attractiveness of extremely high mean lifetimes. Although it may seem silly to spend time and money changing a $MTTF$ from 6 years (50,000 hours) to 45 years (390,000 hours) or 95 years (830,000 hours) because most products are obsolete in less than 10 years, these changes increase the probability of surviving the first 10 years from 0.17 to 0.80 or 0.90, respectively.

Exponential lifetimes admit a simple approximation, $R(t) \approx 1 - t/M$, as long as the time period, t, is small relative to the mean lifetime, M. Another useful way to look at this approximation is

$$\text{Prob("death" before } t) \;=\; 1 - R(t) \;\approx\; t/M \qquad (4)$$

which means that doubling the mean lifetime, M, halves the chance of ‘‘death’’ in time periods, t, that are small relative to M.

The failure rate of a lifetime with an exponential distribution is the reciprocal of mean lifetime, $1/M$. Failure rate is a metric often preferred over that of mean lifetime because of the $R(t) \approx 1 - t/M$ approximation. In this work we will use whichever of these is most intuitive to the matter at hand.

2. Related Work

The reliability of computer systems is a widely researched field. But because our goals in this paper are specific to the reliability of disk arrays only, we will not attempt a complete review of it here. An excellent treatment of the field can be found in the book by Siewiorek and Swarz [27]. In addition to presenting design methodologies and detailed case studies, this book contains a practical discussion of frequently used mathematical techniques including the basic ones we employ in this paper. Briefer treatments of the reliability of computer systems can be found in Avižienis’s classic survey [2] and Nelson’s up-to-date overview [16]. More detailed understanding of the modelling mathematics can be found in a variety of textbooks [4, 21, 28] and survey articles [8].

One result important to understanding the reliability of disk arrays relates to the distribution of the time until failure of a system with redundant parts and dynamic repair. In this case, the system fails when too many components fail before repair can be completed. Arthurs and Stuck develop intuition for this distribution in a paper modelling the reliability of a machine with a single backup machine and a dedicated repairman [1]. They show that the distribution of the time until both machines are concurrently being repaired approaches the exponential distribution as the probability that one machine will fail before the other is repaired approaches zero. Moreover, they show that this is true regardless of the distributions of the time until machine failure and of the time until repair is complete. Their result can be generalized to apply to any system experiencing short periods (e.g. repair) during which it is vulnerable to improbable events [29].

The result obtained by Arthurs and Stuck is important for our research because it is applicable to repairable, redundant disk arrays. These systems suffer infrequent failures and can be repaired within a small number of hours by replacing the failed component and recovering any affected data. With their result, we expect that the time until a disk array suffers a failure causing some data to be unrecoverable – the time until data loss – will be approximately distributed as an exponential random variable. This approximation improves when repair is made faster or failures occur less frequently. Although we have found evidence that this expectation is borne out by our simulations, it should be remembered that our premise is not sensitive to the distributional assumptions we use.

The conventional method for constructing redundant disk systems is based on duplication [11]. The reliability of duplexed, or mirrored, systems and of the closely related Triple-Modular-Redundant (TMR) systems have been well studied [27]. These are both special cases of the disk-array reliability model that we examine in Section 4. Duplexed disks, however, double disk costs. But, by providing two copies from which to select the closer copy, they can also improve performance [5, 6, 7]. To extend these performance advantages, rather than to enhance reliability, some researchers have suggested disk systems with more than two copies of every disk [5, 14] and, although costly, this is available in Digital Equipment Corporation's disk subsystem products [3]. Because the cost of duplicating data is prohibitive for most systems, we will concentrate on the less expensive N+1-parity encoding for redundancy in disk arrays.

In an early paper on repairable, redundant disk arrays, Park and Balasubramanian present an optimistic estimate for the mean time until data is lost [18] – optimistic because their model underestimates the period of time that a disk array is vulnerable during a disk repair. Section 4 examines a more appropriate model for N+1-parity disk arrays that was first applied to disk arrays by Patterson, Gibson, and Katz [19]. Although Park and Balasubramanian discuss both failures in the hardware that supports disks and the inclusion of on-line spare disks, they do not model the effect these factors have on disk array reliability. Sections 5 and 6 in this paper do, however, model the effects on reliability of each of these factors, respectively, and Section 7 examines their combined effects. These sections show that although failures in disk-support hardware can drastically reduce the reliability provided by redundant data, these effects can be substantially overcome. We have previously presented preliminary [9] and exhaustive [10] analysis of these factors.

Ng has studied Section 4's model for reliability in a disk array and extensions for including on-line spare disks [17]. By employing a Markov model simulation tool, he determined that there is little benefit from including more than one spare disk in a disk array of a single parity group of up to 32 disks. In Section 6, we present an analytic model that substantiates and extends his result.

3. Tools and Methods

In our quest to quantify reliability, we developed and used three different types of stochastic tools. The most general tool is an event-driven simulator we wrote called RELI. RELI explicitly generates failure events in a specified disk array from its installation until the first time a failed disk's data cannot be recovered. Each of the durations from installation until data loss is a sample lifetime of the specified disk array. RELI samples enough lifetimes to be able to estimate a reasonably narrow confidence interval for the expected lifetime of the specified disk array.

The second tool we used was a software package called Sharpe that solves Markov models. (Sharpe is distributed by Duke University [22, 23].) It generates a cumulative distribution function for array lifetimes from a completely specified Markov model. To provide results without the trouble of developing complex programs, our final tools are fully-parameterized analytic expressions. These expressions can be evaluated with simple programs or hand-held calculators.

The greatest advantage of simulation is that it allows complexity to be modelled realistically, although complex simulators are vulnerable to lurking bugs. Markov models are well-understood (and the more widely-used tool, Sharpe, can be expected to have fewer bugs), but the specification of a complex Markov model frequently requires many arguable approximations, and its specification task is nearly as difficult as coding a simulator. Furthermore, the solution of Markov models by a tool such as Sharpe requires all parameters to be fully specified, and it generates numeric results. For highly reliable designs, moreover, Sharpe's general solution technique suffers from numerical approximation problems, and it becomes necessary to resort to separate solutions for each point in time of interest. For these reasons we use Markov models mainly to explain and verify simulation and analytic models.

In contrast to either Markov models or simulation models, analytic models offer simpler computation and greater intuitiveness. Unfortunately, they offer point estimates only and usually rely on arguable simplifications. A major goal of this paper was the development of analytic models to obtain these computational and intuitional advantages, but we rely on more complex simulation models to support and extend assumptions in analytic models.

4. Independent Disk Failures

Figure 3 shows the simplest model for single-erasure-correcting arrays of redundant disks. This three-state Markov model has independent and exponential disk lifetimes and exponential repair durations. This is a specific application of a well-understood model that has been featured in stochastic-process textbooks and which exemplifies simple redundancy in a collection of identical, repairable equipment [4]. For this model the mean time until a group suffers data loss, $MTTDL$, is

$$MTTDL = \frac{(2N+1)\lambda+\mu}{N(N+1)\lambda^2} . \qquad (5)$$

The mean time until a group suffers data loss is simply the expected time beginning in state 0 and ending on the transition into state 2 in Figure 3.

Because $\mu = 1/MTTR_{disk}$ is much larger than $\lambda = 1/MTTF_{disk}$, the reliability function, $R(t)$, is well-approximated by an exponential function, $R_{Indep}(t) = e^{-t/MTTDL}$, with the same mean, $MTTDL$. As to the value of $MTTDL$, another approximation is appropriate. Because $(2N+1)\lambda = (2N+1)/MTTF_{disk}$ should be much less than $1/MTTR_{disk} = \mu$, the mean time until data is lost can be approximated as

$$MTTDL_{Indep} = \frac{\mu}{N(N+1)\lambda^2} = \frac{MTTF_{disk}{}^2}{N(N+1)MTTR_{disk}} \ . \tag{6}$$

This simplification is pessimistic by less than 6% for a randomly selected collection of 94 parameter sets (sets of values for $MTTF_{disk}$, $MTTR_{disk}$, and N). Together, these two approximations are

$$R_{Indep}(t) = e^{-t/MTTDL_{Indep}} \ . \tag{7}$$

In practice, a disk array is composed of more than a single parity group. If each group fails independently, the time until data loss in a multiple-group disk array is the time until the first component group fails. Given that the lifetime of a single-erasure-correcting group can be modelled as an exponential random variable, the lifetime of a disk array has the same distribution as the shortest lifetime of its component groups. Conveniently, one of the properties of exponential random variables is that the minimum of multiple exponential random variables is also an exponential random variable whose rate is the sum of the component groups' rates. Hence, a disk array composed of G single-erasure-correcting groups each containing $N+1$ disks has

$$MTTDL_{Indep} = \frac{(2N+1)\lambda+\mu}{GN(N+1)\lambda^2} = \frac{(2N+1)MTTF_{disk}MTTR_{disk}+MTTF_{disk}{}^2}{GN(N+1)MTTR_{disk}} \ , \tag{8}$$

or

$$MTTDL_{Indep} \approx \frac{\mu}{GN(N+1)\lambda^2} = \frac{MTTF_{disk}{}^2}{GN(N+1)MTTR_{disk}} \ , \tag{9}$$

and

$$R_{Indep}(t) = e^{-t/MTTDL_{Indep}} \ . \tag{10}$$

These last two expressions have previously been presented as a model of the reliability of disk arrays [19]. These expressions are optimistic because they neglect the dependent failure modes modelled in the next section. Because they are optimistic, they are best used to demonstrate that a disk array design is at best unreliable rather than to inspire confidence that a particular design is adequately reliable.

Figure 4 shows how 10-year reliability in arrays degrades with increasing numbers of disks and decreasing parity overhead. In this figure a pessimistic value of 24 hours is used for mean disk repair time. This value corresponds roughly to repairing a disk by first having a replacement disk shipped to you in the next day's mail. Small arrays attain high 10-year reliabilities with small overhead for redundant disks, but large arrays require a large overhead to achieve a 10-year reliability much better than a single disk. By comparison, however, the 10-

year reliability of 10 data disks with no redundancy is less than 0.003; that is, there is a 0.3% chance that 10 disks with exponential lifetimes having a mean of 150,000 hours will run for 10 years without any failure. There is virtually no chance of 100 or 1000 of these disks surviving 10 years without loss of data.

5. Dependent Disk Failures

In the previous section we calculated the reliability of a disk array based on the optimistic assumption that all failures are independent. However, most I/O subsystems require support hardware that is shared by multiple disks. For example, power supplies, cabling, cooling, and controllers are often shared across multiple disks. Figure 5 shows an example of such support hardware and their failure rates for commonly available components that might be used in a disk array. This figure assumes that the disks that share cabling also share power and cooling. We call such a configuration a *string.* Strings may fail if any of the support hardware fails, but, for purposes of this analysis, not because of disk failures. String failures can render many disks unavailable. In these cases multiple disks cannot be said to fail in an independent manner. Since redundancy groups based on parity only guarantee recovery of a single disk failure, dependent disk failures may defeat our redundancy scheme.

String failures can severely limit the reliability of a disk array because each failure may render data unavailable for a sufficiently long period that this data is declared effectively lost.[2] Assuming that the time until a string fails is exponentially distributed, string failures cause the rate of data loss to be larger than that of the previous section by up to the rate of string failures ($G_{string} / MTTF_{string}$),

$$\frac{1}{MTTDL_{RAID}} = \frac{G\ (N+1)\,(N+2)\,MTTR_{disk}}{(MTTF_{disk})^2} + \frac{G_{string}}{MTTF_{string}} \tag{11}$$

where G_{string} is the number of strings. String failures limit $MTTDL$ to a maximum of $MTTF_{string} / G_{string}$, regardless of redundancy among the disks.

The standard approach for limiting loss of data caused by string failures is to duplicate power, cooling, and controller components so that $MTTF_{string}$ is maximized. Although full duplication substantially improves the reliability of strings, it is an expensive solution that reduces the frequency of, but does not tolerate, string failures. A more powerful solution capitalizes on the larger number of identical components in a disk array.

[2] There are a variety of reasons for assuming that string failures ‘‘erase’’ the data on their disks. First of all, some string failure, such as power failure, increase the probability that each disk will not restart when power is next applied. More significantly, applications such as on-line transaction processing may stand to lose many times the value of their computer systems when data is unavailable. However, in situations where string failures are non-threatening, the model in the next section will be more appropriate.

With smaller and more numerous disks, an array is likely to contain a sufficient number of strings so that parity groups can be organized with no more than one disk from each group on any one string. As shown in Figure 6, this *orthogonal* organization of strings and parity groups guarantees that a single string failure can be endured as long as no other disk or string failure occurs before the string is repaired.

Repairing a failed string is more complex than repairing an independently failed disk. It involves a service visit or replacement operation for the component of the string that failed and then the recovery of multiple disks. This latter step may be necessary because the disks damaged or rendered unavailable by string failure have been replaced during repair of the string. It may also be necessary because the contents of these disks have been outdated by changes applied to parity disks instead of to the unavailable string-failed disks. Because of the multiple disk recovery step, the array may lose data if other components fail before the slowest disk recovery is complete.

Modelling the reliability of an orthogonal disk array with dependent failures is more complex than the independent disk-failures model of Section 4. Parity groups cannot be modelled individually because string failures cause all groups to experience a failure simultaneously. We have developed a monolithic Markov model for a disk array with G groups organized orthogonally. As before, this model yields disk array lifetime distributions that are well-modelled by exponential random variables with the appropriate mean lifetime. A complete solution for this model's reliability can be computed by standard Markov methods, but the computations depend on G and require messy inverse Laplace transformations. What would be better, then, is a method for deriving simpler estimates of $MTTDL$ directly.

One approach to directly estimating $MTTDL$ begins by recognizing that failures, particularly closely-spaced double failures, are rare in real disk arrays. Where failure events are rare, we assume that different types of data loss are mutually exclusive and sum their rates. There are two sources of data loss in an orthogonal disk array: component failures during the repair of a disk and component failures during the repair of a string. Figures 7a and 7b describe relatively simple sub-models for these two sources of data loss, respectively. Considering each of these cases separately, summing their easily computed contributions and inverting this sum, the mean time to data loss, $MTTDL_{Ortho}$, is

$$\frac{\dfrac{MTTF_{disk}{}^2}{GN(N+1)MTTR_{disk}}}{\dfrac{1+\alpha_F}{1+(2N+1)\varepsilon_{dd}+N\varepsilon_{ds}}+\alpha_F\dfrac{1+\alpha_F\phi/G+(1+\alpha_F/G)(\alpha_R/\alpha_{Rd}+GN\varepsilon_{sd}+(N+1)\phi\varepsilon_{ss})}{((N+1)/MTTF_{string}+(2N+1)\varepsilon_{ss}+GN\varepsilon_{sd})\Psi(N+1)+\Psi(N)}}$$

$$\text{where } \Psi(g)=\alpha_{Rd}+GN\varepsilon_{dd}+g\phi\varepsilon_{ds} \; ,$$

$$\alpha_F=\frac{MTTF_{disk}}{MTTF_{string}} \; , \; \alpha_R=\frac{MTTR_{disk}}{MTTR_{string}} \; , \; \alpha_{Rd}=\frac{MTTR_{disk}}{MTTR_{disk-recovery}} \; ,$$

$$\varepsilon_{dd} = \frac{MTTR_{disk}}{MTTF_{disk}} \ , \ \varepsilon_{ss} = \frac{MTTR_{string}}{MTTF_{string}} \ , \ \varepsilon_{sd} = \frac{MTTR_{string}}{MTTF_{disk}} \ , \text{ and } \ \varepsilon_{ds} = \frac{MTTR_{disk}}{MTTF_{string}} \ ,$$

$$\text{and } \ \phi = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{G} \ . \qquad (12)$$

We have compared $MTTDL_{Ortho}$ given in Equation 12 to a numeric solution of our monolithic Markov model across 87 randomly-selected parameter sets[3]. In fact, there is only one estimate with more than 4% error and it has the unlikely combination of a 72-hour disk-repair time, and a two-hour string-repair time, with an estimated disk-array lifetime of about 10,000 hours. Using $MTTDL_{Ortho}$ as an estimate of the correct $MTTDL$, reliability can be modelled as

$$R_{Ortho}(t) = e^{-t \ / \ MTTDL_{Ortho}} \ . \qquad (13)$$

Although Equation 12 is accurate, it is certainly not simple! The values of ε_{dd}, ε_{ss}, ε_{sd}, and ε_{ds} are expected to be small because they are ratios of a mean repair duration to a mean time until failure. If each ε is assumed to be zero, $MTTDL_{Ortho}$ simplifies to

$$MTTDL'_{Ortho} = \frac{\dfrac{(MTTF_{disk})^2}{G \ (N+1)\,(N+2)\, MTTR_{disk}}}{\left(1 + \alpha_F \left(1 + \dfrac{1}{\alpha_{Rd}} + \dfrac{1}{\alpha_R}\right) + \dfrac{{\alpha_F}^2}{G}\left(\dfrac{\phi}{\alpha_{Rd}} + \dfrac{1}{\alpha_R}\right) \right)} \ . \qquad (14)$$

However, this expression has a substantially larger relative error. For the same 87 parameter sets used to evaluate $MTTDL_{Ortho}$, this expression differs from a numeric solution by more than 15% in 10 cases. While in some cases the simplicity of this expression makes it preferrable to Equation 12, we will not use this estimate further. The well-versed reader may note that an earlier version of this expression [25, 9] neglected to differentiate the disk-repair time after a string repair from the isolated, independent disk-repair time.

We will now adapt the expression for $MTTDL_{Ortho}$, the mean lifetime of an orthogonal disk array, for a more realistic, non-exponential disk-repair time composed of two parts. The first part of a disk repair process is a fixed delivery time at the end of which the failed disk is replaced.[4] Following this replacement, the second part of disk repair is an exponential disk-recovery time. For example, if a particular parameter set in the last section had an average disk-repair time of 24 hours, we might allocate 20 hours to a fixed-length delivery time and then set

[3] To numericly solve Markov models, we use the Sharpe reliability and performance evaluation software package developed at Duke University [23]. Originally, we selected at random 99 parameter sets, but Sharpe failed to evaluate 12 of these sets because of they exceeded the program's limits on the number of states in a Markov model.

[4] Calling a fixed-length delivery time more realistic is perhaps a poor choice of words; in fact, delivery does not take the same amount of time on each occasion, but it is substantially less variable than an exponential random variable. Recovery, on the other hand, will sometimes occur while the disk array is idle, and other times, it will occur during peak user load. If user requests are given priority, then recovery time will be highly variable, a much better match to an exponential random variable.

the average disk-recovery time to four hours. Our RELI simulator implements this split-phase repair process. During a string repair all affected disks are replaced so that disk repair after string repair does not involve any replacement-delivery time. The simulator also assumes that if a disk fails while another failed disk's replacement is awaiting delivery, the second replacement can be added to the existing order and arrive with the first replacement disk. This is a good model for replacements delivered by a repairperson because service personnel are likely to bring a few extra disks when they visit customers. This optimization causes the average replacement-delivery time experienced by a failed disk repair process to be shorter than the actual fixed-length delivery time.

To account for replacement-delivery times that are shortened because disk failures happen close together, we examine each replacement delivery process. Most disk failures initiate an order for a replacement disk because there is no outstanding order. During the fixed-length period, D, until a replacement arrives, each of the other $G(N+1)-1$ disks may fail. Because each other disk fails independently with probability $1 - e^{-D/MTTF_{disk}}$, the expected number of additional failures is $(G(N+1)-1)(1-e^{-D/MTTF_{disk}})$. If the number of disks in the array, $G(N+1)$, is large relative to the number of disks expected to fail during a delivery, then subsequent failures form a Poisson process, and their arrivals are uniformly distributed over the delivery period [20]. For this case the average time a failed disk waits until it is replaced is half the actual fixed-length delivery time. Including the delivery-initiating failure, the average delivery time is

$$\begin{aligned}\text{Average delivery time} &= \frac{\text{Expected total delivery time}}{\text{Expected total failures}} \\ &= \frac{D+(G(N+1)-1)(1-e^{-D/MTTF_{disk}})D/2}{1+(G(N+1)-1)(1-e^{-D/MTTF_{disk}})}\ . \end{aligned} \tag{15}$$

To incorporate these calculations into our estimate, $MTTDL_{Ortho}$, given in Equation 12, we set $MTTR_{disk-recovery}$ to the mean disk-recovery time and set $MTTR_{disk}$ to the average delivery time plus the mean disk-recovery time. Although there is wider variation between simulated and estimated $MTTDL$ with this modified repair time model than with an exponential repair time, in the vast majority of comparisons we found differences of less than ± 5%.

Applying the orthogonal estimate for the mean lifetime of a disk array to our strawman example, we show the impact of string repair and disk delivery durations on data reliability in Figures 8a and 8b. These figures demonstrate two significant characteristics of orthogonal disk arrays. First, average string repair as slow as two weeks does not provide reliability at least as good as a single disk. Second, even if string repair is relatively fast, the delivery time of replacement disks must still be minimized to achieve high reliability. Fast repair processes can be expensive if they call for immediate attention of qualified service personnel. A small pool of on-line spare disks can effectively provide much faster repair without immediate attention from service personnel by reducing disk-delivery time to zero in most cases. The next section estimates the effect of a pool of on-line spares on the reliability of a disk array when string failures do not affect the integrity of disks' data. Then Section 7 estimates the reliability of a disk array that has on-line spares when string failures do affect data integrity.

6. Independent Disk Failures with On-line Spares

Figures 8a and 8b above suggest that on-line spare disks can significantly improve *MTTDL* and reliability. In this section we reduce average disk-repair time by employing a pool of on-line spare disks, which is not a continuation of the model in the last section because we do not include the effects of dependent disk failures. The combined effects of dependent disk failures and on-line spare disks are addressed in Section 7.

On-line spares reduce average disk-repair time because a failed disk can be replaced with a spare in the time it takes to change the software mappings for the location of the failed-disk's data. In this way disk-repair time is just the time it takes to recover a disk's contents by reading all other disks in a group, computing the exclusive-or, and writing these values to the recovering disk. Because there is no immediate need for a person to insert a new disk to replace a failed one, not only is repair and recovery time reduced, but opportunities for human error are eliminated. But, because on-line spares increase the cost of a disk array, their number will be limited, and disk failures will occasionally be exposed to longer periods of repair when the spare pool is exhausted.

Our disk-array simulator can explicitly maintain a pool of spare disks. It uses a *threshold* parameter to decide when to issue an order for disks to replace spares now acting in place of recently failed disks. It also delivers enough disks to completely replenish the spare pool whenever a replacement order is filled.

In all disk arrays protected with N+1 parity, data will be lost whenever two disks in one group fail before the first has been repaired. The state of the spare pool affects the repair rate, however. While there are spares, repair is fast and loss of data is unlikely, but while there are no spares, repair is slow and loss of data is much more likely. A complete Markov model for this involves a state for each combination of the number of groups recovering and the number of spares available. Such a model has many transition rates, each of which offers an opportunity for modelling error. Even with modern software tools that assist in the construction of a model, more complex models require more specification and are more prone to error. Since closely-spaced double-disk failures and spare-pool depletion are both likely to be rare events of relatively short duration, we expect the distribution of time-to-data loss to be roughly exponential. With this expectation we can model reliability by estimating *MTTDL*. Since sources of data loss are rare, we estimate the reciprocal of *MTTDL* by separately modelling each source of data loss and summing their rates of loss:

$$\frac{1}{MTTDL_{IndepSpares}} = \text{Independent disk–failure data–loss rate}$$

$$+ \text{ Spares–exhausted data–loss rate } . \qquad \textbf{(16)}$$

The data-loss rate while on-line spares are available is given Equation 8 in Section 4 with $MTTR_{disk}$ set to the average time it takes to remap and recover a failed disk onto a spare one, $MTTR_{disk-recovery}$. To estimate the data-loss rate arising because the spare pool periodically empties, we treat each period between the refilling of the spare pool as an independent opportunity to lose data and assume that data loss events occur at the end of one of these periods.[5] Figure 9 shows an example of a sequence of three opportunities for losing data occur during orders. The time until data loss caused by spare-pool exhaustion will be a geometric random variable with mean equal to the average time between spare-pool refilling divided by the probability of data loss during the delivery of an order.

$$\text{Spares–exhausted data–loss rate} = \frac{\text{Probability of data loss per order}}{\text{Average time between filled orders}} \qquad (17)$$

Putting these terms together, the mean lifetime of a disk array suffering only independent disk failures and augmented with an on-line spare pool is $MTTDL_{IndepSpares}$:

$$MTTDL_{IndepSpares} = \qquad (18)$$

$$\frac{1}{\dfrac{GN(N+1)MTTR_{disk}}{((2N+1)MTTR_{disk}+MTTF_{disk})MTTF_{disk}} + \dfrac{\text{P(data loss per order)}}{\text{Average time between filled orders}}} \, ,$$

where $MTTR_{disk}$ is the mean time for disk recovery excluding replacement-disk delivery time.

To compute the probability of data loss per order, we condition on the number of disk failures that occur while an order is being delivered. This number has a binomial distribution based on the probability that a disk will fail in the fixed-length delivery period. The result then follows because computing the probability of data loss given a particular number of failures during an order is a simple matter of counting the number of configurations that have more than one failure in at least one group. After these computations, the probability of data loss during an order becomes

$$\text{P(data loss per order)} = \sum_{q=2}^{G} \left[\begin{matrix} G(N+1)+T \\ T+q \end{matrix}\right] (1-e^{-\lambda D})^{T+q} e^{-\lambda D(G(N+1)-q)} \left[1 - \prod_{i=0}^{q-1} \frac{(G-i)(N+1)}{G(N+1)-i} \right]$$

$$+ \sum_{q=G+1}^{G(N+1)} \left[\begin{matrix} G(N+1)+T \\ T+q \end{matrix}\right] (1-e^{-\lambda D})^{T+q} e^{-\lambda D(G(N+1)-q)} \, . \qquad (19)$$

Finally, the average time between successive refillings of the spare pool is the time it takes to deliver replacement disks plus the expected time for the failure of the number of disks required to diminish the spare pool to its threshold. This latter is the expected time until the first $S-T$ disks have failed beginning with $G(N+1)+S$

[5] This approximation is good as long as D, the delivery time, is small relative to the average time required for $T+G$ disks to fail, where T is the order threshold and G is the number of parity groups.

operational disks.

$$\text{Average time between filled orders} = D + MTTF_{disk} \sum_{j=G(N+1)+T+1}^{G(N+1)+S} \frac{1}{j} \tag{20}$$

We have compared this estimate of disk array mean lifetime, $MTTDL_{IndepSpares}$, to simulation-derived estimates for a randomly selected collection of 100 parameter sets. This estimate was within ± 5% in most cases and within ± 10% in all cases. These simulations also supported our expectation that disk-array lifetimes under this model are also well-modelled with an exponential distribution.

Investigating the effects of spare disks on the mean lifetime of a disk array that does not suffer multiple simultaneous disk failures, we apply $MTTDL_{IndepSpares}$ to our strawman example. Figures 10a and 10b show the $MTTDL$ and the 10-year reliability as a function of the size of an on-line spare pool. In this figure, replacement disks are not ordered until all spares have been assigned to replace failed disks. This reorder policy will amortize the cost of delivering a set of new spare disks over the number of disks in this set. This can yield substantial savings beause field-service visits may cost between one tenth and one half of the cost of a disk [17].

Although the improvement in $MTTDL$ with increasing spare disks is much less with longer disk-delivery times, the 10-year reliability exceeds 0.90 with only one spare disk even if the disk-delivery time is 336 hours (two weeks)! If a replacement disk is ordered as soon as a disk fails instead of after the last spare is assigned to a failure (not shown in Figure 10), then all three $MTTDL$ curves achieve 29,000,000 hours with only four spare disks, and all three reliability curves exceed 0.99 with only two spare disks!

Figure 11 depicts the relationship between the disk-delivery time, the replacement-order threshold, and the maximum number of spares for our strawman disk array. Because an order delivery involves a person meddling with the disk array, a high maximum number of spares and a low reorder threshold reduces the frequency that the disk array is exposed to human error. However, a large spare pool increases disk and support-hardware costs and a low threshold requires faster disk delivery to avoid lowered reliability. This figure displays the maximum number of hours that disk delivery can take without causing the 10-year reliability to drop under 0.995. With a maximum of four spare disks in this array of 70 data disks, a disk delivery time of less than about 100 hours allows this 10-year reliability goal to be met with minimal reorder threshold of zero. For less pressure on disk delivery time, an reorder threshold of one is a good compromise between infrequent orders and inexpensive disk delivery.

These figures show the substantial benefits provided by even a small pool of on-line spares. However, this model fails to consider the negative consequences of string failures demonstrated in Section 5. The next section addresses this problem.

7. Dependent Disk Failures with On-line Spares

This section examines disk array reliability under the influence of dependent disk failures and with the benefit of on-line spare disks. Section 5 shows that dependent disk failures, although tolerable, can dramatically reduce the reliability that would be expected from the same disk array suffering only independent failures. On the other hand, Section 6 shows that a small number of on-line spares can dramatically improve the reliability of a disk array suffering only independent disk failures. Naturally, we would like to use on-line spare disks to overcome the limitations imposed by dependent disk failures.

By including on-line spare disks into an orthogonal array, a few new issues are introduced because on-line disks must be attached to support hardware somewhere in the array. Figure 12 shows how an orthogonal array with spare disks may surrender its orthogonality while waiting for recently failed disks to be replaced. A second issue is the allocation of spare disks to strings. We may choose to allocate spare disks together in one string, as we have done in Figure 12, or spread them out over strings that also contain data or parity disks. For a small number of spares, these two choices yield comparable reliabilities. When the number of spares is equal to or larger than the number of disks attached to a string, however, a string containing only spare disks effectively replaces a failed string without disturbing the array's orthogonality. In contrast, if on-line spare disks are allocated one to each string then the failure of a string may negate orthogonality until the failed string is repaired and its disks's contents recovered. Figure 13 shows an example where a string has failed in an orthogonal array with spare disks allocated one to each string. In this case, there are enough spare disks to immediately replace the failed string, but each spare is assigned into a parity group already represented on the spare's string. Until the failed string is repaired, a second string failure will cause data to be lost even if no recoveries are in progress. Because of this increased vulnerability, we allocate spare disks to strings containing only spare disks for the rest of this paper.

Once again, simulated lifetimes in arrays with on-line spares and string failures are well-modelled by an exponential distribution. This means that a complete reliability model depends only on an estimate for mean disk-array lifetime, $MTTDL$:

$$R(t) = e^{-t/MTTDL} \ . \qquad (21)$$

A complete Markov model for dependent disk failures and on-line spare disks will have an enormous number of states unless clever exploitation of model symmetries can be exploited. Where the state space is large, not only would the task of specifying transition rates be error-prone, but tools like Sharpe would be unable to solve the models. For these reasons we do not present a complete Markov model.

One alternative to developing a complete Markov model is to apply the approach used in Section 6. To do this, we would need to compute the expected time between instances where the disk array is fully populated and properly orthogonal and the probability that data is lost in each of these intervals. At this time, we have not found a model that accurately estimates the mean lifetime of a disk array for each configuration of the spare pool. Instead, we describe the effect that on-line spare disks have on the mean lifetime of a disk array for particular sizes of the spare disk pool. This model, albiet an incomplete one, describes the general form of the mean time until data is lost as a function of the number of spare disks.

By examining a variety of simulation results, we find that some configurations achieve their maximum mean lifetime with only a few spares and other configurations do not approach their maximum mean lifetime until they include many spares. However, increasing the number of spare disks beyond one string of spare disks does not significantly improve mean lifetime. This observation, that one string populated with on-line spare disks appears to be all that is needed to maximize the mean lifetime of a disk array, is the basis of this section's model, and is illustrated in Figure 14. It estimates $MTTDL$ when there are zero, one, two, and an infinite number of strings populated with on-line spare disks. In most cases, one string of spare disks yields as high an $MTTDL$ as two strings of spare disks, and in almost all cases two strings of spare disks yield as high an $MTTDL$ as infinitely many strings of spare disks.

7.1. Infinite Spares Bound

First, we present an upper bound on $MTTDL$ (and, consequently, reliability) in a disk array that suffers both independent disk failures and string failures but that has the benefit of on-line spare disks. This upper bound models the case where there are infinitely many strings fully populated with spare disks. As in Section 5, the sources of data loss are secondary failures during an individual disk repair or an individual string repair. Because concurrent double failures are rare, we separately model these two cases and estimate the overall rate at which data is lost as the sum of the rates of data loss arising from each of these component models. Figures 15a and 15b show sub-models for data losses initiated by individual disk failures and individual string failures, respectively. Applying the methods used in Section 4 and 5, a bound on the mean lifetime of a disk array of G parity groups, each containing $N+1$ disks plus an infinite number of spare disks and strings, is:

$$MTTDL_{InfiniteSpares} = \frac{\dfrac{MTTF_{disk}^{\,2}}{GN(N+1)MTTR_{disk-recovery}}}{\dfrac{1+\alpha_F}{1+(2N+1)\varepsilon'_{dd}+N\varepsilon'_{ds}} + \dfrac{\alpha_F(1+\alpha_F\phi/G)}{1+GN\varepsilon'_{dd}+(2N+1)\varepsilon'_{ds}}} \qquad (22)$$

$$\text{where } \alpha_F = \frac{MTTF_{disk}}{MTTF_{string}},\ \varepsilon'_{dd} = \frac{MTTR_{disk-recovery}}{MTTF_{disk}},\ \varepsilon'_{ds} = \frac{MTTR_{disk-recovery}}{MTTF_{string}},$$

$$\text{and } \phi = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{G} \, .$$

Mean array lifetime closely approaches this bound with at most two strings of spare disks and more often with only one string. The next sections present estimates of mean array lifetime with one and two strings of spare disks and contrast these estimates to this infinite spare disks bound.

7.2. One and Two Strings of On-Line Spare Disks

Next, we present an estimate for the mean lifetime of an orthogonal disk array with one and two strings of on-line spare disks. Once again we estimate the contribution to the overall rate at which data is lost from each source of loss separately and then sum these estimates. In the case of one string of on-line spare disks, we consider three submodels for data loss: the infinite-spare-disks model from Equation 22 in the last section, the independent-disk-failures-with-spare-disks model from Equation 18 in Section 6, and a new model that accounts for data losses triggered by the failure of one or two more strings during a string repair. The mean lifetime of an orthogonal disk array with one string of spare disks, $MTTDL_{OneSpareString}$, is then:

$$\frac{1}{MTTDL_{OneSpareString}} = \text{Infinite–spares data–loss rate} + \text{Spares–exhausted data–loss rate}$$
$$+ \text{ 1Spare–string–repairing data–loss rate} \, . \qquad (23)$$

Figure 16 describes the sub-model for the data-loss rate arising from additional string failures during string repair. Using our methods for deriving the mean time until data is lost beginning with an extra string of spare disks, the third component of the rate of data loss in this section is:

$$\text{1Spare–string–repairing data–loss rate} = \frac{\lambda_1(\lambda_2\lambda_3+\lambda_3\lambda_4+\lambda_4\mu_2)}{\lambda_1(\lambda_2+\lambda_3)+\lambda_3(\lambda_2+\lambda_4+\mu_1)+\mu_2(\lambda_1+\lambda_4+\mu_1)}$$

$$\text{where} \quad \lambda_1 = \frac{(N+2)}{MTTF_{string}} \, , \;\; \lambda_2 = \frac{(N+1)(1-\delta)}{MTTF_{string}} \, , \;\; \lambda_3 = \frac{N}{MTTF_{string}} + \frac{GN}{MTTF_{disk}} \, ,$$

$$\lambda_4 = \frac{(N+1)\delta}{MTTF_{string}} + \frac{N\,\delta'}{MTTF_{disk}} \, , \;\; \mu_1 = \frac{1}{MTTR_{string}} \, , \;\; \mu_2 = \frac{2}{MTTR_{string}} \, ,$$

$$\delta = 1 - \pi_0 \, , \qquad \text{and} \qquad \delta' = \sum_{i=0}^{G} i\,\pi_i \, ,$$

where π_i, $i = 1, 2, ..., G$, is given by:

$$\pi_i = \binom{G}{i} \left[\frac{(N+1)\,\overline{D}}{MTTF_{disk}} \right]^i \pi_0 \, , \quad \text{and} \quad \pi_0 = \left\{ \sum_{i=0}^{G} \binom{G}{i} \left[\frac{(N+1)\,\overline{D}}{MTTF_{disk}} \right]^i \right\}^{-1} . \qquad (24)$$

In this model we use two parameters, δ and δ′, to approximate more complex interactions between string failures and the replacement of failed data disks. The parameter δ represents the probability that during the repair of a string failure there are also failed and unspared disks on other unfailed strings. Similarily, the parameter δ′ represents the expected number of these unspared failed disks when a second string fails during the repair of a first failed string. To estimate δ and δ', we have constructed a more complex Markov model whose state records the number of failed and unspared disks during the repair of string failures and solved for the steady state probabilities, π_i [10]. Our solution assumes that the threshold for ordering replacement disks is one less than the maximum number of spares; that is, an order for a replacement is issued immediately after each failure (unless there is already an outstanding order). In practise, this assumption may not always be desirable, but it is not unreasonable because immediate reorder maximizes mean lifetime and reliability.

We have compared this model's estimates for mean time until data loss to estimates made by simulation and found agreement up to the inherent variation of simulation sampling. As we have done in earlier sections, this comparison is made for 100 parameter sets selected at random from a large set of conceivable values. The selected parameter sets are not intended to represent typical choices; in fact, they are intended to stress reasonable choices for parameters to test that the models are accurate for a range of choices from poor to good. Therefore, we do not claim that for these 100 parameter sets the relative difference between the infinite-spares estimate for $MTTDL$ in Equation 22 and this single-string-of-spare-disks estimate is representative of their relative difference in a set of good choices for disk array parameters. Nevertheless, Figure 17 shows their relative difference in this collection of 100 parameter sets. The infinite-spares estimate is more than 10% larger than the single-string-of-spare-disks estimate in 19 of the 100 parameter sets. This suggests that in some cases there is a potential for significant benefit from more than one string of spare disks.

Finally, our model for mean time until data is lost in a disk array with two strings of on-line spare disks is a straightforward extension of these calculations:

$$\frac{1}{MTTDL_{TwoSpareStrings}} = \text{Infinite–spares data–loss rate} + \text{Spares–exhausted data–loss rate}$$

$$+ \text{2Spare–string–repairing data–loss rate} \,. \qquad (25)$$

Applying the same methods,

$$\text{2Spare–string–repairing data–loss rate} = \qquad (26)$$

$$\frac{\lambda_1\lambda_2(\lambda_3\lambda_4+\lambda_4\lambda_5+\lambda_5\mu_3)}{\lambda_1\lambda_2(\lambda_3+\lambda_4)+\lambda_4(\lambda_1+\lambda_2)(\lambda_3+\lambda_5)+\lambda_4(\mu_1(\lambda_3+\lambda_5)+\mu_2(\lambda_1+\mu_1))+\mu_3(\lambda_1(\lambda_2+\mu_2)+\lambda_5(\lambda_1+\lambda_2+\mu_1))+\mu_1\mu_2\mu_3}$$

$$\text{where} \quad \lambda_1 = \frac{(N+3)}{MTTF_{string}} \,,\; \lambda_2 = \frac{(N+2)}{MTTF_{string}} \,,\; \lambda_3 = \frac{(N+1)(1-\delta)}{MTTF_{string}} \,,$$

$$\lambda_4 = \frac{N}{MTTF_{string}} + \frac{GN}{MTTF_{disk}} \,,\; \lambda_5 = \frac{(N+1)\delta}{MTTF_{string}} + \frac{N\delta'}{MTTF_{disk}} \,,$$

$$\mu_1 = \frac{1}{MTTR_{string}} \,,\; \mu_2 = \frac{2}{MTTR_{string}} \,,\; \text{and} \;\; \mu_3 = \frac{3}{MTTR_{string}} \,.$$

Again, this estimate for mean lifetime in a disk array with two strings of spare disks is in good agreement with simulation's estimates. Moreover, except for one of the 100 parameter sets evaluated, two strings of spare disks yields a mean lifetime within 1% of the mean lifetime in an array with an infinite number of spare disks and strings. Even the one parameter set that has a relative difference larger than 1% only differs by 15%. Based on the simulation results in this section and the last section, the mean lifetime of an array with an infinite number of spare disks and strings is frequently achieved with one string of spare disks and is almost always achieved with two strings of spare disks.

7.3. Designing Disk Arrays with On-Line Spares and String Failures

In this section we present analysis of four issues important to the design of our strawman disk array when it employs on-line spare disks:

(1) using spare disks with large parity groups to achieve higher reliability than is provided by arrays with a higher fraction of redundant disks,

(2) reducing disk-recovery time to dramatically improve mean disk-array lifetime in arrays with one or two strings of on-line spare disks,

(3) determining limits on reliability benefits provided by adding redundancy to disk-support hardware, and

(4) examining reliability when strings of spare disks are partially populated and replacement-disk reordering is not done immediately after each failure.

We evaluate the first three of these issues using the models developed in the last sections. Because the fourth of these issues does not meet the assumptions of these models, we explore it with simulation results.

All four of these issues are explored using the context of the strawman disk array we introduced in Table 1 of the introduction to this paper. Unless we explicitly vary a parameter, our strawman disk array has seven parity groups with 10 data disks and a parity disk in each. Each disk and each string has an expontentially distributed lifetime with a mean of 150,000 hours. Disk recovery and string repair also have exponentially distribued durations with means one hour and 72 hours, respectively. Replacement-disk delivery time is the minimum of 72 hours or the time until an already ordered replacement disk arrives.

7.3.1. Higher Overhead for Redundancy May Not Improve Reliability

More redundancy should yield higher reliability. For example, the simple disk array lifetime model given in Equations 8 and 9 of Section 4 shows that *MTTDL* is inversely proportional to the number of disks in a parity group; larger parity groups have lower overhead and lower reliability. In this model, a disk array with mirrored disks is more reliabile than a disk array with N+1-parity redundancy. But the array with mirroring contains up to twice as many disks so its higher reliability is achieved at a substantial cost. This section shows that a disk array with N+1-parity redundancy and on-line spare disks can provide higher reliability at lower cost than a mirrored disk array with the same amount of user data.

Figure 18 shows that our strawman disk array with on-line spare disks can achieve better reliability than a comparable mirrored disk array. With only one string of spare disks, a N+1-parity disk array with a 72-hour replacement-disk delivery time is more reliable than a mirrored disk array with either a 72- or an 18-hour replacement-disk delivery time. Even if the mirrored disk’s replacement-disk delivery time is reduced to 4 hours it is still less reliable than our strawman disk array with a 72 hour replacement-disk delivery time and one string of spare disks unless mean string-repair time in both is less than seven hours. Additionally, our strawman disk array with one string of spare disks, a 72 hour replacement-disk delivery time, and a 72 hour mean string-repair time is as reliable as a mirrored disk array with a four hour replacement-disk delivery time and a 11 hour mean string-repair time. Because reducing disk-replacement and string-repair time requires increased availability of expensive human service, the cost advantage of our strawman disk array is even better than is suggested by the comparison of its 84 disks to the mirrored disk array’s 140 disks!

The reliability advantages of N+1 parity are even better if two strings of spare disks are included. In this case our strawman disk array still has a 91 disks to 140 disks cost advantage over the mirrored disk array and its reliability is insensitive to mean string repair times as large as two weeks. Unless the mirrored disk array has a replacement-disk delivery time of four hours or less and a mean string-repair time of less than seven hours, our strawman disk array with a replacement-disk delivery time of 72 hours has better reliability.

Figure 18b shows the probability that these N+1-parity and mirrored disk arrays survive 10 years of operation without data loss. All configuration have a better than 75% chance of surviving 10 years and, if the mean string-repair time is less than 37 hours, all configurations have a better than 90% chance of surviving 10 years. These 10-year reliabilities are substantially higher than the 56% chance that a single disk with a 150,000 hour mean lifetime survives 10 years.

Because four of the five configurations shown in Figure 18b have almost the same 10-year reliability, there seems little reason other than cost to prefer one configuration over others. In particular, the relatively large differences in mean lifetime shown in Figure 18a do not appear in Figure 18b. These differences in mean lifetime are significant, however, if we consider the fraction of disk arrays that suffer data loss in 10 years. Equation 4 in Section 1 shows that doubling the mean lifetime of a disk array will halve the expected number of ‘‘angry’’ customers even though the probability that an individual disk array survives all 10 years without data loss is only increased a small amount. For example, 2.2% of all mirrored disk arrays with a replacement-disk delivery time of four hours and a mean string-repair time of 25 hours will lose data in 10 years, but only 1.0% of all N+1-parity disk arrays with two strings of spare disks, a replacement-disk delivery time of 72 hours, and a mean string-repair time of 72 hours, will lose data in 10 years.

These figures show the superior reliability of N+1-parity disk arrays with spare disks in comparison to mirrored disk arrays without spare disks. This is a reasonable comparison because N+1-parity disk arrays are less expensive than mirrored disk arrays even with spare disks. If comparable numbers of spare disks are added to mirrored disk arrays, increasing the cost differential, then mirrored disks achieve superior reliability.

7.3.2. Higher Reliability Through Faster Disk Recovery

To recover the contents of a failed disk in an N+1-parity disk array, all remaining disks in the failed disk's parity group must be entirely read and the failed disk's replacement must be entirely written. Before a block can be written to the failed disk's replacement, the corresponding blocks from each of the rest of the disks in the parity group must be have been read and their collective parity (exclusive-or) must be computed. This collective parity is exactly the failed disk's missing data, so the collective parity is then written to the failed disk's replacement disk.

If the array controller or host computer managing a failed disk's recovery has sufficient control, transfer, and exclusive-or bandwidth to read all remaining disks and write the replacement disk in parallel, a failed disk's recovery can be completed in about six minutes [26]. This maximum recovery rate is rarely attained because many systems are not designed with sufficient bandwidth. Even if high speed recovery is possible, it would block all user accesses into the entire parity group for the duration of the recovery. In many computer systems, the unavailability of user data for many minutes or tens of minutes is tantamount to data loss because (1) the unavailable data may be out-of-date before it again supports user accesses or (2) the financial penalties derived from stalling accesses until recovery is complete are unacceptably high. Systems with these kind of high availability requirements may demand that user accesses be served during disk recovery. This will cause a failed disk's recovery be slowed down. Because the failed disk's data can be recovered block-by-block in any order, user accesses for any data in the effected parity group can be serviced, albiet at reduced performance [15].

Figure 19 shows the effect of changing the mean disk-recovery time on the mean disk-array lifetime. If there are no spares in an N+1-parity disk array, then the array's vulnerability to data loss is largely determined by replacement-disk delivery time. In this case, disk recovery can be slowed to one or four hours on average to accomodate user accesses without significant effect on reliability. If, instead, there are two strings of spare disks in the array, the array's vulnerability to data loss is largely determined by disk-recovery time. In this case, increasing the mean disk-recovery time by a factor of 10 from six minutes to an hour reduces mean lifetime by a factor of 10 which, in turn, increases the expected fraction of disk arrays that will lose data by the same factor of 10. With just one string of spare disks this effect is less pronounced, but it is still important for high reliability to minimize disk-recovery time.

7.3.3. Higher Array Reliability Through Higher String Reliability

As we mentioned in Figure 5 of Section 5, the conventional approach for avoiding low reliability in disk-support hardware is to employ more reliable, and more expensive, parts. For even higher string reliability, at even higher costs, support-hardware components can be made redundant. Because the fraction of a disk array's cost that is attributable to support-hardware is not likely to be large, it is reasonable to evaluate the contribution to array reliability that results from increased string reliability via higher quality parts or redundancy.

Figure 20 shows the effect of varying string reliability on the mean lifetime of our strawman disk array. If the unenhanced mean string lifetime is low – in this example, less than 100,000 hours – then doubling it doubles the mean lifetime. This effect is more pronounced in arrays that have one or more strings of spare disks. When mean string lifetime approaches or exceeds 1,000,000 hours, however, increasing it further provides little benefit for array reliability. Unless the unenhanced string reliability is very low and the cost of increasing mean string lifetime by a factor of 100 or 1,000 increases array cost by less than 10%, adding a string of spare disks is a more effective method of increasing reliability than is increasing mean string lifetime.

7.3.4. Partially Populated Spare Strings and Low Reorder Thresholds

In contrast to figures in the last three sections, this section presents simulation data instead of model estimates. It investigates aspects of the reliability of our strawman disk array that do not meet the assumptions of our models. In particular, this section explores two design alternatives:

(1) One or two extra strings partially populated with spare disks are desirable because disks are expensive.

(2) A reorder threshold that does not cause an order to be issued immediately after every disk failure is desirable because it reduces the frequency that service personnel interacts with the disk array and because it delays the purchase of new disks.

Figures 21a and 21b show simulated estimates of mean lifetime and 10-year reliability, respectively, for our strawman disk array. Figure 21b shows that incorporating a single spare disk increases the chance that an individual disk array will survive 10 years without data loss from 52% to 80% and that incorporating one string of spare disks increases the chance of surviving 10 years without data loss to over 90% for all three reorder thresholds. Toward the basic goal of providing better reliability than a single disk drive, which has a 56% chance of surviving 10 years without failure, these results indicate that a small pool of spare disks with any reorder threshold is all that is necessary.

Figure 21a addresses high reliability in our strawman disk array. This figure shows that an immediate reorder policy achieves maximum reliability levels with substantially fewer spare disks than the delayed reorder policies. It also shows that high reliability is not achieved with any less than a fully populated string of spare disks.

Intuitively, we understand this figure by estimating the average number of spare disks in the spare pool. Because replacement-disk delivery time is much shorter than the expected time until the next disk failure, the average number of spare disks on hand is about half way between the maximum number of spare disks and one more than the threshold. This means that with immediate reorder, the array can nearly always immediately replace all disks on a failed string. This also explains why a half-empty reorder policy achieves the reliability of one string of spare disks with an immediate reorder policy once it has about one and a half strings of spare disks. This intuition is not satisfactory for explaining the reliability of the empty reorder policy because it incorrectly suggests that two strings of spare disks would be sufficient to achieve the reliability of one string of spare disks with an immediate reorder policy, which is inaccurate.

The benefit of a delayed reorder policy is largely derived from a reduction in the frequency that service personnel interacts with the disk array. The average rate of these interactions is the average rate of disk failures divided by the number of disks that must fail before replacement disks are reordered.

Average human interactions per hour = (27)

$$\frac{G(N+2)+(S+T+1)/2}{S-T}\left[\frac{1}{MTTF_{disk}}+\frac{1}{MTTF_{string}}\right].$$

When our strawman disk array has one string of spare disks and an immediate reorder policy ($S=7, T=6$), this rate is 182/150,000. When it has one and a half strings of spare disks and a half empty reorder policy ($S=11, T=6$), this rate is 186/(5×150,000), nearly five times lower without loss of reliability! Finally, when it has two strings of spare disks and an empty reorder policy ($S=14, T=0$), this rate is 184/(14×150,000), 14 times lower than when the array has one string of spare disks and an immediate reorder policy. Unfortunately, the reliability of this third case is also substantially lower.

The results in Figure 21a indicate that our strawman disk array needs at least one string of spare disks to achieve high reliability. With four more spare disks and a policy of reordering spare disks when the pool is half empty, the frequency of human interaction with the array can be reduced by a factor of five without sacrificing reliability.

8. Summary and Conclusions

Throughout this paper we have used a strawman disk array to exemplify our models and explore array design issues. Table 1 in this paper's introduction presents the strawman disk array as an attractive alternative to IBM's top-end disk subsystem, the IBM 3390. Table 2 summarizes estimates for the strawman disk array's mean time until data is lost and its 1-, 3-, and 10-year reliability that were presented in Sections 4, 5, 6, and 7. This table shows that without redundancy the strawman disk array has virtually no chance of surviving three or more years without data loss. This is the primary reason for including redundancy in a disk array. Table 2 also shows that, in addition to overcoming this basic threat to data reliability, redundancy can provide high reliability with low overhead costs.

In the data of Table 2, we have used 10% overhead for the parity redundancy. This level of protection more than compensates for threats to reliability from independent failures alone and nearly compensates for the threats to reliability from independent and dependent disk failures without requiring on-line spare disks. If the only threats to data reliability are from independent disk failures, high reliability is achieved with only one on-line spare disk. In this case two on-line spare disks are approximately as useful as an infinite number of on-line spares. Where dependent disk failures also threaten data reliability, Section 7.3.4 shows that just one on-line spare disk achieves higher reliability than provided by a single disk and that high reliability is provided by one string fully populated with on-line spare disks. In this case, two strings populated with on-line spare disks provides as high reliability as an infinite number of strings populated with on-line spare disks.

Section 7.3 examines the design of our strawman disk array in greater detail. It shows, in Section 7.3.1, that with one string of on-line spare disks, the strawman disk array achieves higher reliability than a comparable collection of mirrored disks at lower cost and with less expensive repair processes. Section 7.3.2 then looks at the effect of varying disk recovery time that would result from varying the priority of recovery relative to normal user accesses. It finds that without on-line spare disks, reliability is insensitive to changes in the disk recovery rate, but with on-line spare disks, slowing the disk recovery rate substantially decreases the mean time until data is lost. The next section, 7.3.3, shows that adding on-line spare disks is generally more effective for improving the mean time until data is lost than is improving individual string reliability. Finally, Section 7.3.4 examines the possibility of reducing costs and opportunities for human error by delaying replacement-disk reordering until the spare pool is half full. It finds that with one and half strings of on-line spare disks, the mean time until data is lost in an array with a half full reorder policy is comparable to an array with an immediate reorder policy and one string of on-line spare disks. This means that the frequency of replenishing the spare pool can be reduced by a factor of five at a cost increase of less than 5% of the cost of the non-redundant array.

The net implication of this paper's reliability models is that, subject to our assumptions, our strawman disk array can be made more reliable than a mirrored IBM 3390 disk subsystem at a cost less than one IBM 3390!

To conclude, this paper evaluates the reliability of disk arrays that employ N+1-parity redundancy to tolerate catastrophic failures. Because arrays of small diameter disks contain many more components than the large diameter disks they replace, their non-redundant reliability is unacceptably low. This is the primary need for redundancy; to insure that disk arrays are at least as reliable as the single disks they replace. Secondarily, many owners of computer systems have much higher reliability requirments for their storage systems. These customers have traditionally doubled their expenditures for magnetic disks and duplicated all of their data. Redundant disk arrays offer the opportunity to provide such customers the high reliability they seek at a much lower cost.

In this paper we present models for disk array reliability and their implications for the design of these arrays. These models are analytic expressions based on Markov models of each source of data loss. They account for dependent disk failures, such as support-hardware failures that effect multiple disks, as well as independent disk failures. They also incorporate the benefits of on-line spare disks. These models have been validated against a detailed disk-array lifetime simulator for a wide variety of parameter selections.

The models we present in this paper show that a redundant disk array can easily be designed to provide higher reliability than a single disk. Moreover, with a small overhead for parity and spare disks, a redundant disk array can achieve very high reliability. For some configurations including our strawman configuration, a N+1-parity disk array with on-line spare disks achieves higher reliability than the more expensive mirrored disk array.

As more and more reliability is required of more and more general purpose computer systems, reliability-cost tradeoffs will become critical. The models and design implications developed in this paper will enable secondary storage system designers to achieve reliability goals with cost-effective redundant disk array solutions.

9. Acknowledgements

The contents of this paper grew out of research in Martin Schulze‘s masters degree report and became a chapter in Gibson's Ph.D. dissertation. In developing the models in this paper, we relied heavily on the experience and insight of Ronald Wolff.

This paper also benefited substantially from the efforts of four others, Tom Philippi, Barbara Beach, Terri Watson, and Daniel Stodolsky. Their careful proofreading, extensive editorial suggestions, and gentle humour made our task much more bearable. Daniel's careful study was especially rewarding because he discovered and helped correct significant mistakes in both text and models.

Finally, the benefits of working in the midst of the enriching environment provided by Berkeley's Redundant Arrays of Inexpensive Disks (RAID) project are difficult to measure. Many of its members, notably Randy Katz, Ken Lutz, Peter Chen, Ed Lee, Ann Chervenak, Rich Drewes, Ethan Miller, and Martin Schulze, contributed broadly to our understanding of disk arrays.

10. References

(1) Arthurs, E., and Stuck, B. W., ''A Theoretical Reliability Analysis of a Single Machine with One Cold Standby Machine and One Repairman,'' *Proceedings of the 8th International Symposium on Computer Performance Modeling, Measurement, and Evaluation (PERFORMANCE '81),* F. J. Kylstra (Ed.), North-Holland, November 1981, pp 479-488.

(2) Avižienis, A., ''Fault-Tolerance: The Survival Attribute of Digital Systems,'' *Proceedings of the IEEE,* Volume 66 (10), October 1978, pp 1109-1125.

(3) Bates, K. H., ''Performance Aspects of the HSC Controller,'' *Digital Technical Journal,* Volume 8, February 1989.

(4) Bhat, U. N., *Elements of Applied Stochastic Processes,* John Wiley & Sons, 1972.

(5) Bitton, D., and Gray, J., ''Disk Shadowing,'' *Proceedings of the 14th International Conference on Very Large Data Bases (VLDB),* 1988, pp 331-338.

(6) Chen, P. M., ''An Evaluation of Redundant Arrays of Disks Using an Amdahl 5890,'' University of California Technical Report UCB/CSD 89/506, Berkeley CA, May 1989, Master's Thesis.

(7) Chen, P. M., Gibson, G. A., Katz, R. H., and Patterson, D. A., ''An Evaluation of Redundant Arrays of Disks Using an Amdahl 5890,'' *Proceedings of the 1990 ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS),* Boulder CO, May 1990.

(8) Geist, R., and Trivedi, K., ''Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques,'' *IEEE Computer,* Volume 23 (7), July 1990, pp 52-61.

(9) Gibson, G. A., ''Reliability and Performance in Redundant Arrays of Magnetic Disks,'' *International Conference on Management and Performance Evaluation of Computer Systems (CMG) XX Proceedings,* Computer Measurement Group, Chicago IL, December 1989.

(10) Gibson, G. A., *Redundant Disk Arrays: Reliable, Parallel Secondary Storage,* PhD Dissertation, Technical Report UCB/CSD 91/613, University of California, Berkeley, May 1991. Also to be published in the ACM Distinguished Dissertation series by MIT Press.

(11) Katzman, J. A., ''System Architecture for Nonstop Computing,'' *14th IEEE Computer Society International Conference (COMPCON 77),* February 1977.

(12) Kim, M. Y., and Patel, A. M., ''Error-Correcting Codes for Interleaved Disks with Minimal Redundancy,'' IBM Computer Science Research Report, RC11185 (50403), May 1985.

(13) Livny, M., Khoshafian, S., and Boral, H., ‘‘Multi-disk Management Algorithms,’’ *Proceedings of the 1987 ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS),* May 1987.

(14) Matloff, N. S., ‘‘A Multiple-Disk System for Both Fault Tolerance and Improved Performance,’’ *IEEE Transaction on Reliability,* Volume R-36 (2), June 1987, pp 199-201.

(15) Muntz, R. R., and Lui, J. C. S., ‘‘Performance Analysis of Disk Arrays Under Failure,’’ *Proceedings of the 16th International Conference on Very Large Data Bases (VLDB),* Dennis McLeod, Ron Sacks-Davis, Hans Schek (Eds.), Morgan Kaufmann Publishers, August 1990, pp 162-173.

(16) Nelson, V. P., ‘‘Fault-Tolerant Computing: Fundamental Concepts,’’ *IEEE Computer*, Volume 23 (7), July 1990, pp 19-25.

(17) Ng, S. W., ‘‘Sparing for Redundant Disk Arrays,’’ IBM Almaden Research Lab presentation, July 1990.

(18) Park, A., and Balasubramanian, K., ‘‘Providing Fault Tolerance in Parallel Secondary Storage Systems,’’ Princeton Technical Report CS-TR-057-86, November 1986.

(19) Patterson, D. A., Gibson, G. A., and Katz, R. H., ‘‘A Case for Redundant Arrays of Inexpensive Disks (RAID),’’ *Proceedings of the 1988 ACM Conference on Management of Data (SIGMOD),* Chicago IL, June 1988, pp 109-116.

(20) Ross, S. M., *Stochastic Processes,* Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, 1983.

(21) Ross, S. M., *Introduction to Probability Models, Third Edition,* Academic Press, Harcourt Brace Jovanovich, 1985.

(22) Sahner, R. A., and Trivedi, K. S., ‘‘Sharpe: Symbolic Hierarchical Automated Reliability and Performance Evaluator, Introduction and Guide for Users,’’ Department of Computer Science, Duke University, September 1986.

(23) Sahner, R. A., and Trivedi, K. S., ‘‘Reliability Modeling using SHARPE,’’ *IEEE Transactions on Reliability,* Volume R-36 (2), June 1987, pp 186-193.

(24) Salem, K., and Garcia-Molina, H., ‘‘Disk Striping,’’ *Proceedings of the 2nd IEEE International Conference on Data Engineering,* 1986.

(25) Schulze, M. E., Gibson, G. A., Katz, R. H., and Patterson, D. A., ‘‘How Reliable is a RAID?’’ *Proceedings of the 1989 IEEE Computer Society International Conference (COMPCON 89),* San Francisco CA, Spring 1989.

(26) Sierra, H. M., *An Introduction to Direct Access Storage Devices,* Academic Press, 1990.

(27) Siewiorek, D. P., and Swarz, R. S., *The Theory and Practice of Reliable System Design*, Digital Press, 1982.

(28) Wolff, R. W., *Stochastic Modeling and the Theory of Queues,* Prentice Hall, 1989.

(29) Wolff, R. W., personal communications, January 1991.

11. Biographies

Garth A. Gibson received the B.Math degree in computer science and applied mathematics from the University of Waterloo, Canada, in 1983, and the M.S. and Ph.D. degrees in computer science from the University of California, Berkeley, in 1987 and 1991, respectively. From May 1991 until August 1992 he was a Research Computer Scientist in the School of Computer Science at Carnegie Mellon University. Since September 1992 he has held an Assistant Professor appointment at CMU in both the School of Computer Science and in the Department of Electrical and Computer Engineering. Dr. Gibson also participates in the NSF sponsored Data Storage Systems Center (DSSC) where he is the Laboratory Director for Storage and Computer Systems Integration. Dr. Gibson’s dissertation was awarded second place in the 1991 ACM Doctoral Dissertation competition.

David A. Patterson joined the faculty at the University of California, Berkeley, after receiving his doctorate from UCLA in 1976. He spent the next several years leading projects at Berkeley involving Reduced Instruction Set Computers. These projects were the basis of the SPARC architecture used by several companies. He also spent a leave at Thinking Machines Corporation where he made contributions to the CM-5. He is the co-author of four books, and consults for several computer companies. Patterson received the Distinguished Teaching Award from the University of California, the Karlstrom Outstanding Educator Award from the ACM, and was named an IEEE Fellow. He is currently chair of the CS Division in the EECS Department at Berkeley.

Metric	IBM 3390	IBM 0661
Units	1	70
Formatted Data Capacity (MB)	22700	22400
Number of Actuators	12	70
Avg Access Time (msec)	19.7	19.8
Max I/Os/Sec/Box	609	3535
Track Transfer Rate (MB/sec)	15.3	118.3

Table 1: Comparison of Strawman Disk Array to IBM 3390. *An array of 70 IBM 0661 disks is used in this paper to exemplify reliability models. This ‘‘strawman’’ was selected to match the capacity of an IBM 3390 disk subsystem, IBM’s high-end disk product. This table shows that the strawman array has superior throughput and comparable response time. Relative price is not as clear, but, using 1991 typical costs, 70 3½-inch disks will cost a disk array manufacturer about 22,400 MB × 2.5 $/MB = $56,000 whereas IBM’s best customers must pay almost three times this price for an IBM 3390 and a portion of an IBM 3990 controller. Notice that this comparison neglects the cost of an array controller and other array support hardware as well as the inflation from cost to price.*

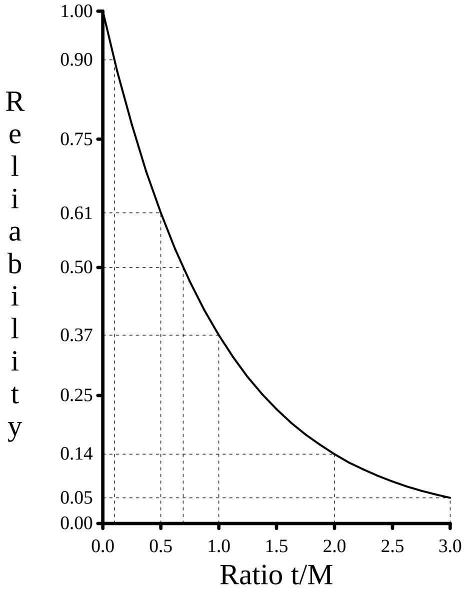


Figure 1: Exponential Reliability versus Ratio of Time to Mean Lifetime. *This figure shows the reliability of a system with exponential lifetimes with time expressed as a multiple of the system's mean lifetime, M. We have marked a few interesting points; the system has a 90% chance of surviving 0.1 M, a 61% chance of surviving 0.5 M, a 37% chance of surviving 1.0 M, a 14% chance of surviving 2.0 M, and a 5% chance of surviving 3.0 M. We have also marked the median lifetime, the time yielding a 50% chance of survival, which is 0.69 M.*

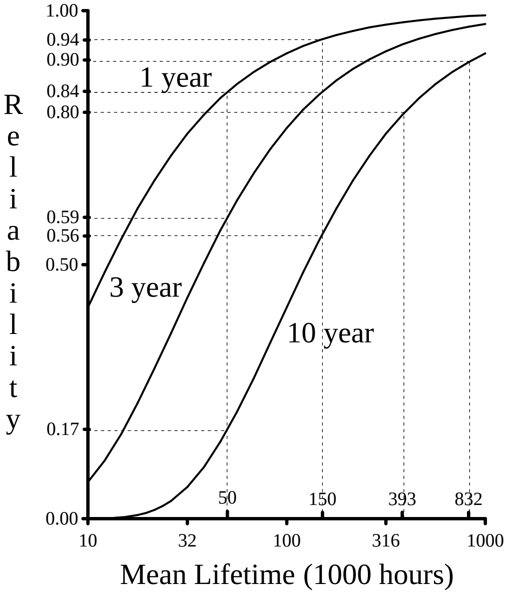


Figure 2: Exponential 1-, 3-, and 10-Year Reliabilities versus Mean Lifetime. *This figure shows the reliability of a system with exponential lifetimes over 1, 3, and 10 years as a function of the system's mean lifetime (in 1,000 hours, where there are 8,766 hours in a year). The x-axis scale is logarithmic. With a mean lifetime of 50,000 hours, the system has an 84% chance of surviving one year, a 59% chance of surviving three years, and a 17% chance of surviving 10 years. If the mean lifetime can be increased to 150,000 hours, the chance of surviving one year rises to 94%, the chance of surviving three years rises to 84%, and the chance of surviving 10 years rises to 56%. To achieve an 80% or 90% chance of surviving 10 years, the mean lifetime must exceed 390,000 hours or 830,000 hours, respectively.*

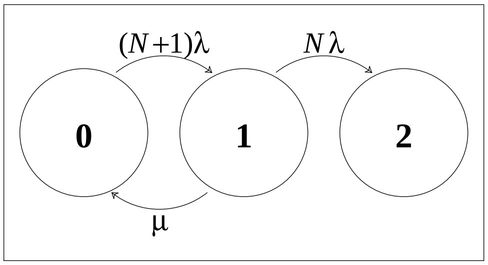


Figure 3: Data Loss Model for Independent Disk Failures in a Single Parity Group. *A single disk-array parity-group of $N+1$ disks can be modelled by a three-state Markov model if disk lifetimes are exponential with mean $MTTF_{disk} = 1/\lambda$ and disk repairs are exponential with mean $MTTR_{disk} = 1/\mu$. The states are labeled with the number of disk failures evidenced by the array. When there are no failures, the rate of failure is $N+1$ times the rate of an individual disk failure, λ. When there is one failure, the rate of repair is μ and the rate of a second failure is $N\lambda$. Once there are two concurrent failures, data has been lost, and there are no more transitions.*

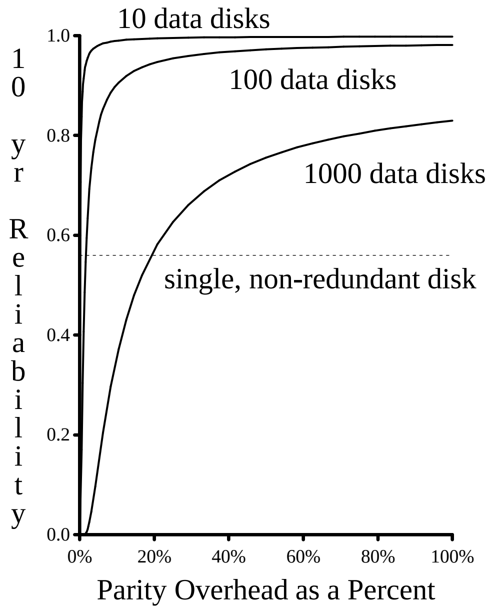


Figure 4: Reliability versus Redundancy Overhead. *This graph shows the trend for the 10-year reliability in a disk array suffering only independent exponential failures when each disk has a mean lifetime of 150,000 hours and it takes an average of 24 hours to repair each disk failure. Three array sizes are shown: 10, 100, and 1000 data disks per array. The size of a parity group,* $N+1$*, is related to the parity overhead expressed as a percent,* O *, by* $N = 100/O$ *Recall from Figure 2 that a single disk with exponential lifetimes and a mean lifetime of 150,000 hours has a 10-year reliability of 0.56 (dotted line). Therefore, even a disk array with 1,000 data disks can be made more reliable than a single disk at about 20% overhead!*

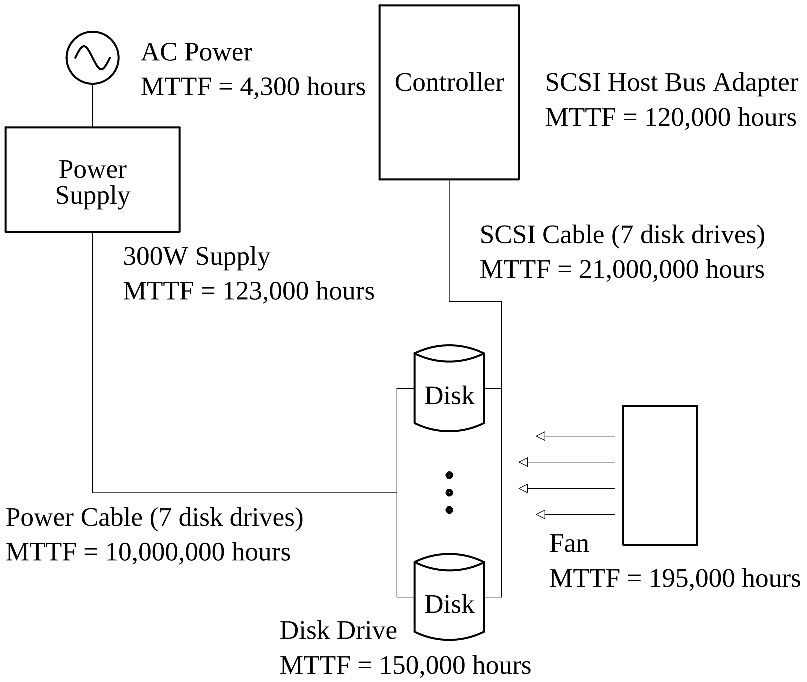


Figure 5: Example of Support Hardware Shared by Multiple Disks. *Disk subsystems are usually partitioned into* strings *that share datapath cabling and controllers. Although it is possible for different collections of disks to share power supplies and cooling support, this figure shows an example of a single string where cabling, controller, power, and cooling are all shared by the same disks. Sample component reliability figures are shown for a relatively low-cost, SCSI interface design [25]. Assuming that the support hardware's components have exponentially distributed lifetimes, the overall failure rate of the* non-disk *portion of a string is the sum of its components' failure rates. For high data reliability it is essential that the external power grid be isolated from the system. Using this data and assuming (possibly battery-backed-up) power supplies uneffected by power grid irregularities, the mean time to failure of a string is about 46,000 hours. This estimate can be easily increased by using more reliable and expensive parts or by incorporating redundant support hardware components, but its low value relative to mean disk lifetimes suggests that dependent disk failures can have a severe effect on reliability. In the rest of this paper we use a more reasonable value of 150,000 hours as the mean time to string failure in our examples of a strawman disk array. The effect of varying string reliability is explored in Section 7.3.3.*

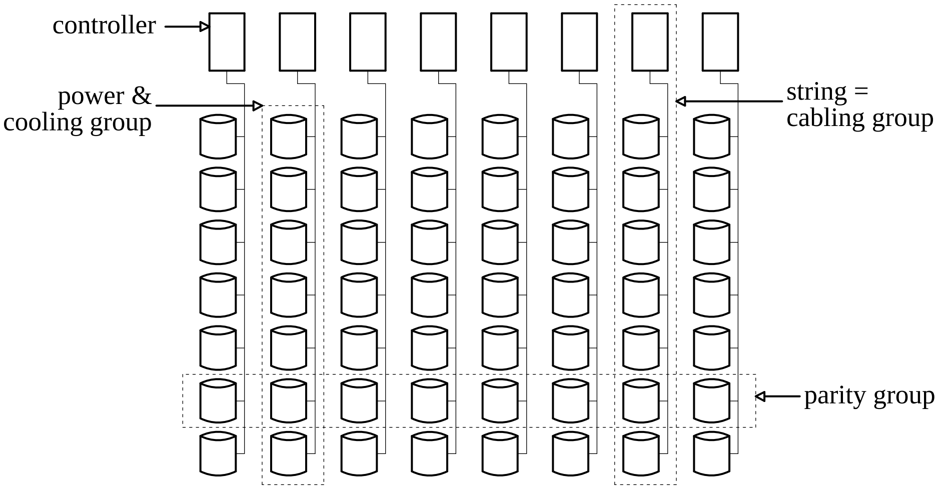
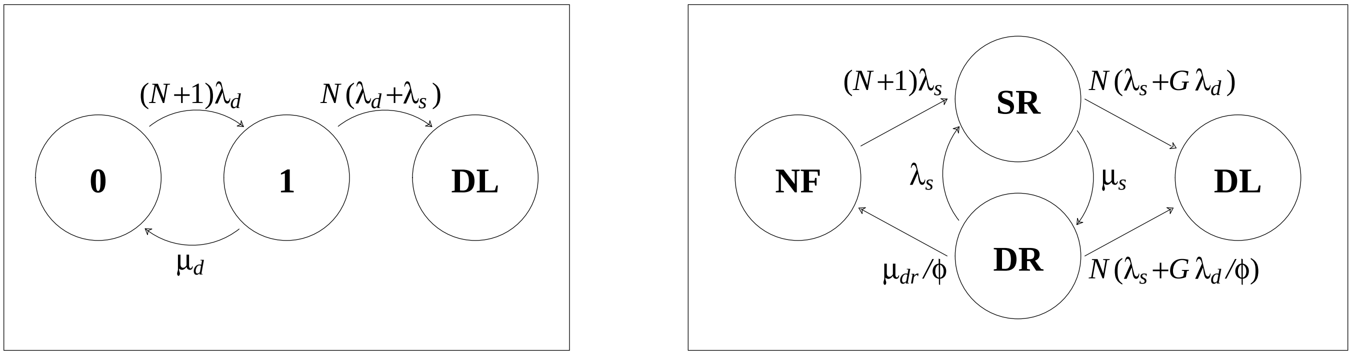
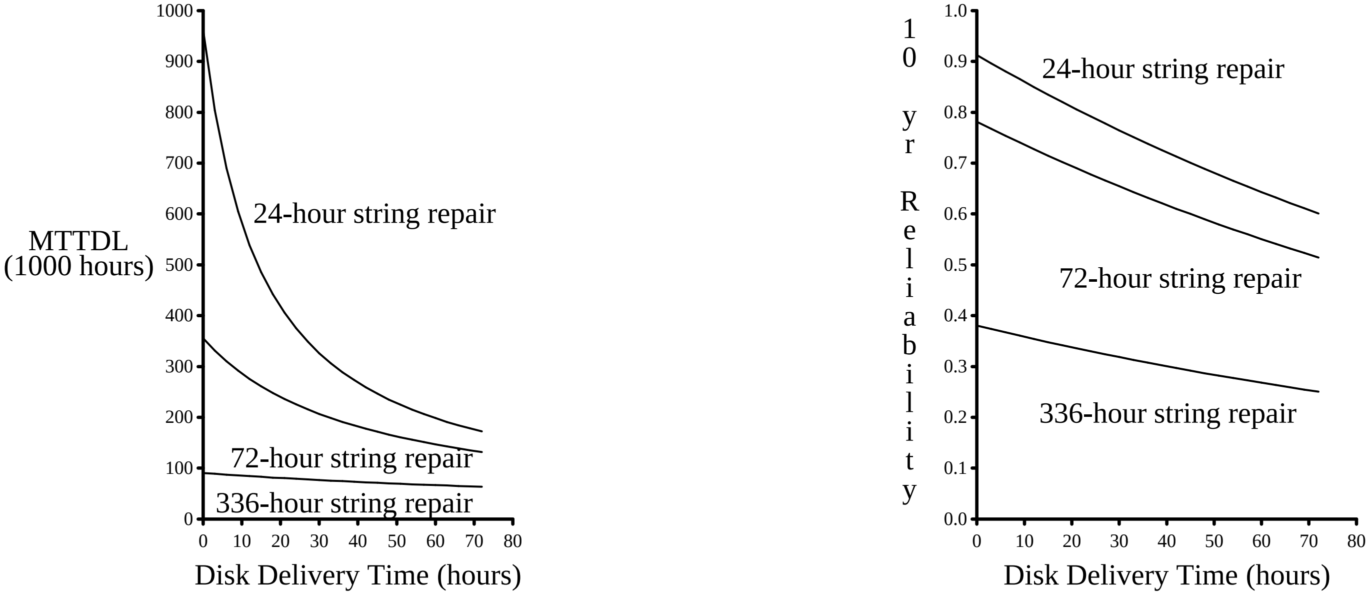


Figure 6: Orthogonal Organization of Parity and Support Hardware Groups. *By organizing support-hardware groups orthogonal to parity groups, the failure of a support-hardware group, or* string*, will destroy at most one disk in each parity group. Since parity-based redundancy schemes handle one failure per group, single-string failures are survivable.*



Figures 7a and 7b: Submodels for Data Loss in Orthogonal Disk Arrays. *The two sources of data loss in orthogonal disk arrays are failures during a disk repair and failures during a string repair. Figure 7a, on the left, shows the sub-model for data loss in a single parity group caused by a second failure during a (non-string-failed) disk repair. Each of the $N+1$ disks in a parity group fails independently with $MTTF_{disk} = 1/\lambda_d$ and is repaired with $MTTR_{disk} = 1/\mu_d$. While a disk is being repaired, the failure of any of the other N disks or their strings causes data loss. (This model is the same as the model of Figure 3 with different transition rates.) Figure 7b, on the right, shows the sub-model for data loss caused by a second failure during a string repair. Each of the $N+1$ strings in the array fails independently with $MTTF_{string} = 1/\lambda_s$ and is repaired with $MTTR_{string} = 1/\mu_s$. During the repair of a string, the failure of any of the other N strings or the remaining GN disks will cause data loss. Once a string is repaired, the data on its disks is recovered either because the repaired string has new disks or because user data has been updated for the repaired string's disks using the array's parity disks. The average recovery time of a disk after a string repair, $MTTR_{disk-recovery} = 1/\mu_{dr}$, may be less than the average disk repair time, $MTTR_{disk} = 1/\mu_d$, because the string repair process is likely to include necessary disk replacements. While the disks of a recently-repaired string are recovering, the renewed failure of the same string reinitiates string repair. To avoid data loss after a string repair all G disks must complete recovery; therefore, the transition rate to state* **NF** *is the reciprocal of the expected maximum of G disk repairs, $MTTR_{Gdisks} = \phi/\mu_{dr}$, where $\phi = 1/1+1/2+...+1/G$. The factor ϕ also modifies the rate of data loss during string-induced disk recovery to account for the reduced vulnerability to other disk failures as indvidual recoveries are completed.*



Figures 8a and 8b: Repair Durations in Orthogonal Strawman. *The MTTDL and 10-year reliability of our strawman disk array are sensitive to their repair durations. Figure 8a shows MTTDL and Figure 8b shows 10-year reliability as they are effected by both average string-repair time and replacement-disk delivery time. The strawman disk array has an orthogonal organization of seven parity groups of 10 data disks plus a parity disk. Disks and strings both have mean lifetimes of 150,000 hours. Average disk-recovery time, excluding replacement-delivery time, is one hour.*

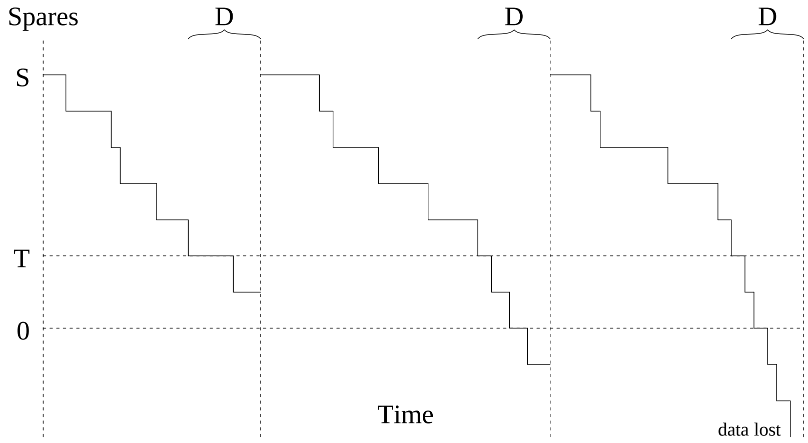
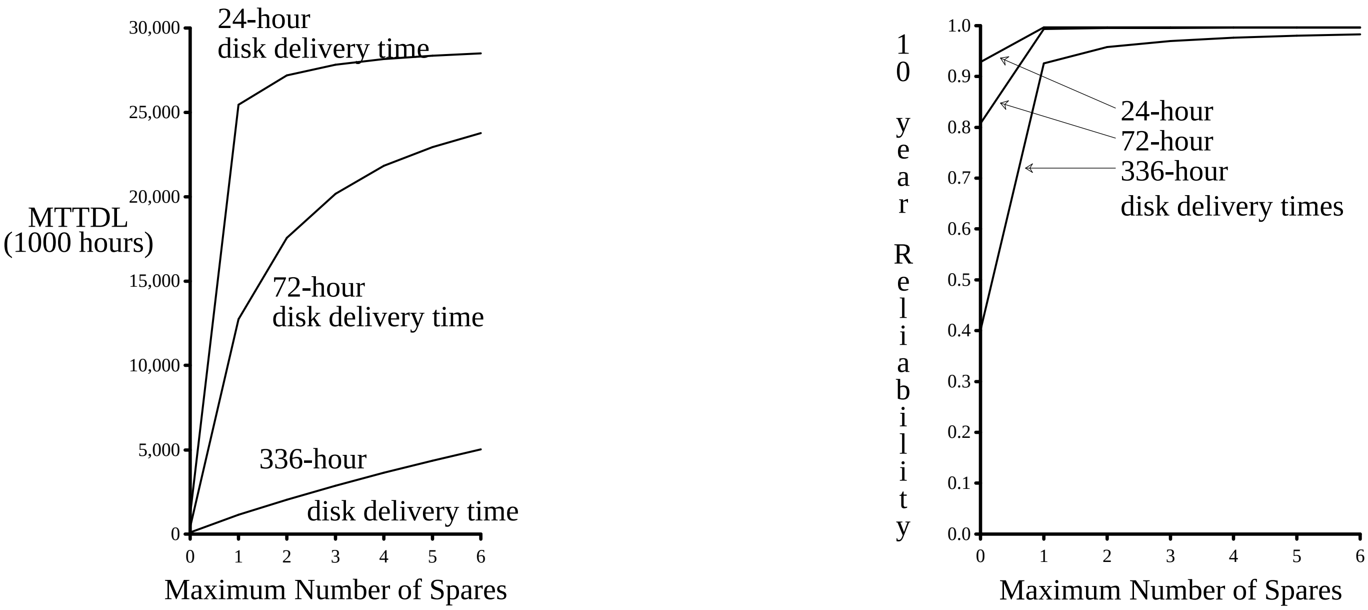


Figure 9: Example of Spare-Pool Depletion and Data Loss. *As disks fail the number of spares decreases from its maximum, S , to a threshold, T, at which time an order for more spares is issued. Orders take time D to be filled and result in a completely refilled spare pool. Until the order is filled disks may continue to fail. Additional failures consume the remaining T spares and then expose the array to a high level of vulnerability. This figure shows three opportunities for data loss during three successive orders. The first does not empty the spare pool, the second empties the spare pool then suffers and survives one additional failure, and the third suffers data loss because too many failures occur before the order is delivered.*



Figures 10a and 10b: Evaluating Benefits of a Small Pool of Spare Disks. *A small pool of spare disks alleviates the effects of disk-delivery time in our strawman disk array if it is not subject to string failures. On the left, Figure 10a shows the MTTDL against the maximum (and initial) number of spare disks. On the right, Figure 10b shows the 10-year reliability for the strawman disk array against the same metric. Recall that this array has seven parity groups of 10 data disks and a parity disks. Each disk has a mean lifetime of 150,000 hours. When a disk fails and a spare is available, disk recovery takes one hour on average. Replacement disks for the spare pool are ordered when the last spare is assigned to replace a failed disk and are delivered in 24 hours (one day), 72 hours (three days) or 336 hours (two weeks).*

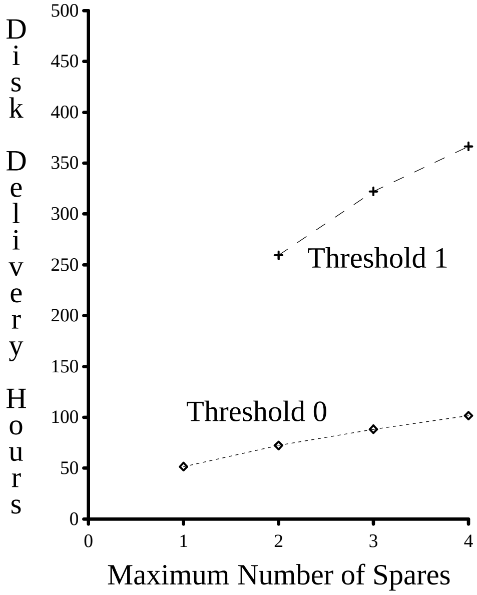


Figure 11: Pool Size, Reorder Threshold, versus Disk Delivery Time. *This figure shows the maximum disk delivery time that yields a 10-year reliability greater than or equal to 0.995 in our strawman disk array. This array has seven parity groups of 10 data disks and a parity disk. Each disk has a mean lifetime of 150,000 hours and a mean recovery time of 1 hour. A low threshold leads to less frequent orders. Although this requires faster disk delivery or larger spare pools, it also reduces the frequency that error-prone humans tamper with the disk array.*

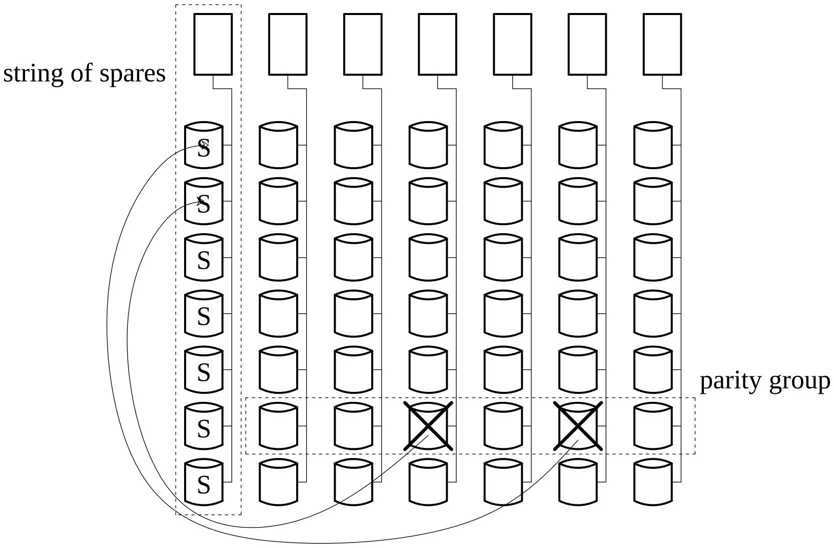


Figure 12: Two Failed and Spared Disks Example. *This example of an orthogonal disk array has suffered two disk failures at different times in the same parity group (N+1=6). In this case the second failure did not happen until after the first failure's assigned spare disk completed recovering the first failure's data. Because the two failure recoveries did not overlap, the disk array remains operational without loss of data. Notice that the two assigned spares are physically in the same string and logically in the same parity group. Until the failed disks are replaced and rebuilt (by copying from their respective spare disks or by another recovery operation), this array is not orthogonal. While it is not orthogonal, it is not protected against loss of data because the isolated failure of the string of spares may erase two disks in one parity group. This period of vulnerability can be reduced, but not eliminated, by manually removing the failed disks, moving two of the unassigned spare disks into the vacated positions, and copying or recoverying the assigned spare disks' contents onto the correctly-located disks. This operation to reduce vulnerability is an unnecessary, and possibly error-prone, human interaction with the disk array; we will not explore this method further in this paper.*

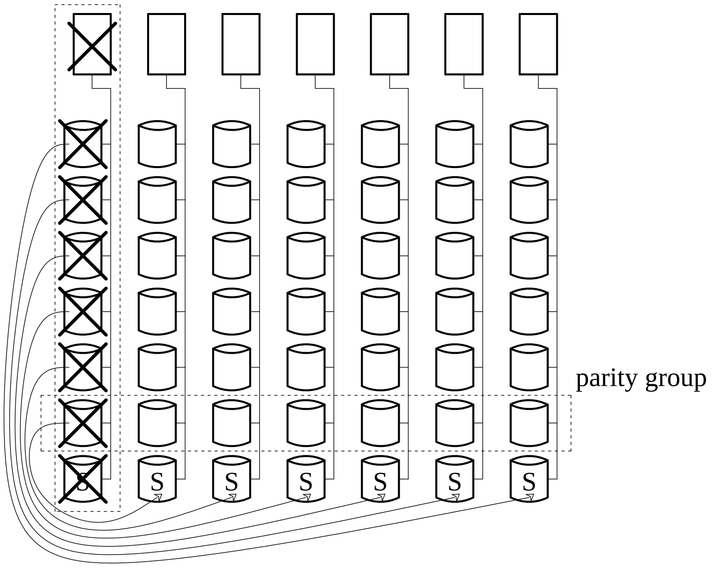


Figure 13: String Failed and Spared Example. *This example shows the effect on orthogonality of a string failure when on-line spare disks are allocated one to a string (unlike the case shown in Figure 12). Although there are enough spares to replace all data and parity disks affected by this string failure, until the failed string is repaired every other string has two disks that are logically in the same parity group.*

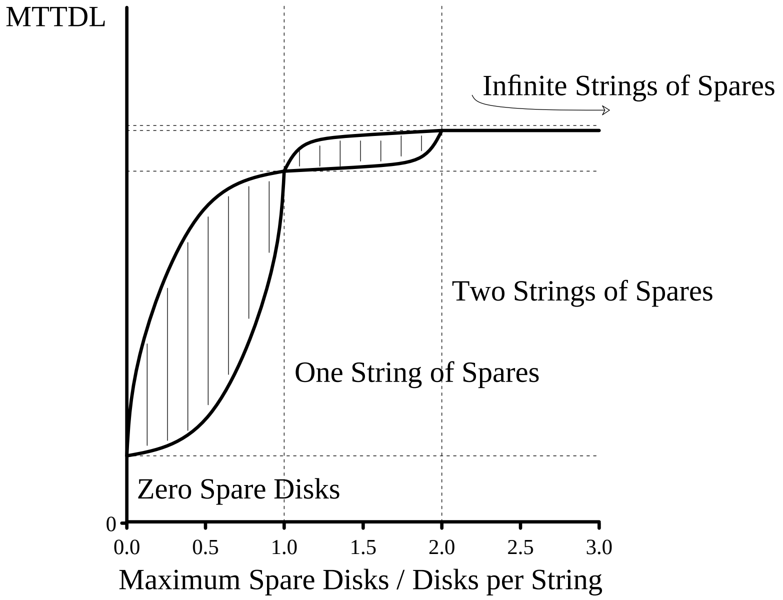
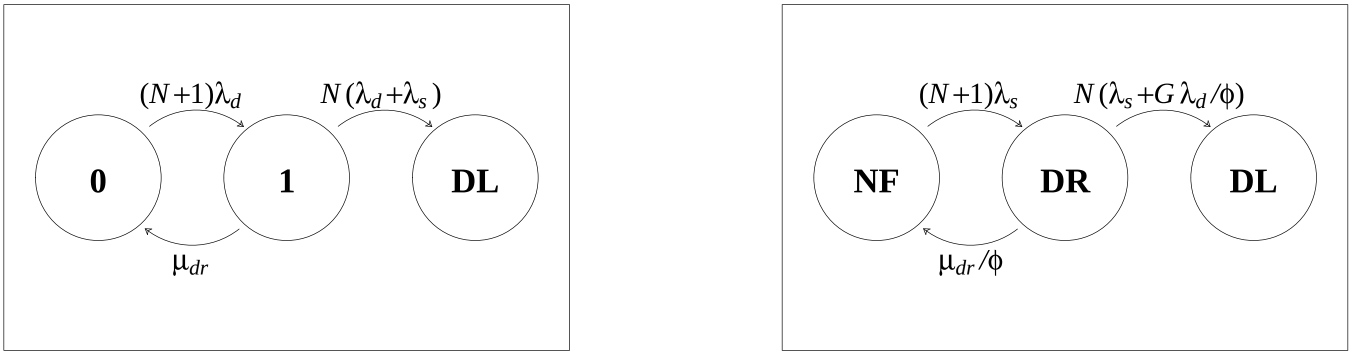


Figure 14: Generic Model for MTTDL. *This figure shows the general form of this section's simple model for the MTTDL of a disk array that suffers string failures as well as individual disk failures but has the benefit of on-line spare disks. This model estimates MTTDL when there are no on-line spare disks, when there is one string populated with on-line spare disks, when there are two strings populated with on-line spare disks, and when there are an infinite number of strings populated with on-line spare disks.*



Figures 15a and 15b: Submodels for Orthogonal Disk Arrays with Infinite Spares. *These models are similar to those in Section 5, Figures 7a and 7b, which model data loss sources in an orthogonal array without on-line spare disks. In both cases the two sources of data loss in orthogonal disk arrays are additional failures during a disk repair and additional failures during a string repair. Figure 15a, on the left, is the same as Figure 7a except that disk replacement is immediate, so disk repair requires recovery only. It shows the submodel for data loss in a single parity group caused by a second failure during a (non-string-failed) disk recovery. Each of the* $N+1$ *disks in a parity group fails independently with* $MTTF_{disk} = 1/\lambda_d$ *and is recovered with* $MTTR_{disk-recovery} = 1/\mu_{dr}$. *While a disk is being repaired, the failure of any of the other* N *disks or their strings causes data loss. There are* G *instances of this sub-model contributing to the overall data loss rate because there are* G *parity groups in the disk array. Figure 15b, like Figure 7b, shows the submodel for data loss caused by a second failure during or soon after a string repair. Because an infinite number of spare disks and strings are available, string replacement is immediate. In this case the only period of vulnerability is the time required to recover of each of the disks on the replaced string. This means that after one of the* $N+1$ *strings, each with* $MTTF_{string} = 1/\lambda_s$, *fails, the array remains vulnerable to data loss on the next failure until all* G *disks on the replacement string have recovered. As in Figure 7b, the rate at which the slowest of* G *disk recoveries takes to complete is the reciprocal of the expected maximum of* G *disk recoveries,* $MTTR_{Gdisks} = \phi/\mu_{dr}$, *where* $\phi = 1/1+1/2+...+1/G$. *While at least one disk is still recovering, data will be lost with the failure of any of the other* N *strings or the failure of any other disk in a parity group that is still recovering. The average number of parity strings vulnerable while at least one disk is still recovering is* G/ϕ.

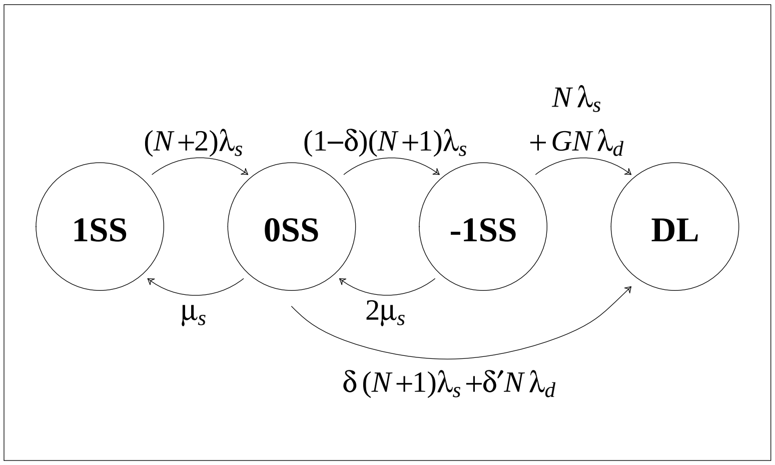


Figure 16: Submodel for Orthogonal Array with One String of Spares. *When there is only one string of spare disks, not all string failures will find a replacement string available immediately. This figure shows a Markov model for data losses that occur while a failed string is forced to wait for a replacement string. While there is one spare string available, state* 1**SS**, *each of the* $N+2$ *strings fails with* $MTTF_{string} = 1/\lambda_s$ *and is repaired or replaced with* $MTTR_{string} = 1/\mu_s$. *If one of the remaining* $N+1$ *strings fails, it causes data loss with probability* δ *because individual disk repairs were in progress. With probability* $1-\delta$, *a second string failure while the first is being repaired is survived. Similarly, data losses can be caused because other disks fail independently in any one of an average of* δ' *groups being repaired. While there are two strings being repaired, state* -1**SS**, *each at rate* $MTTR_{string} = 1/\mu_s$, *any other string failure or individual disk failure on another string will cause data loss.*

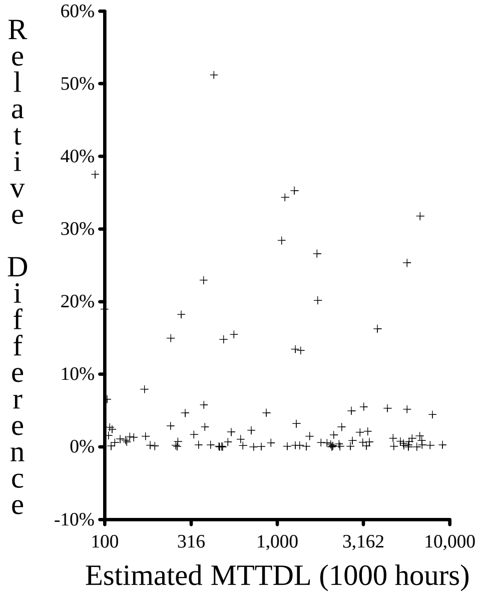
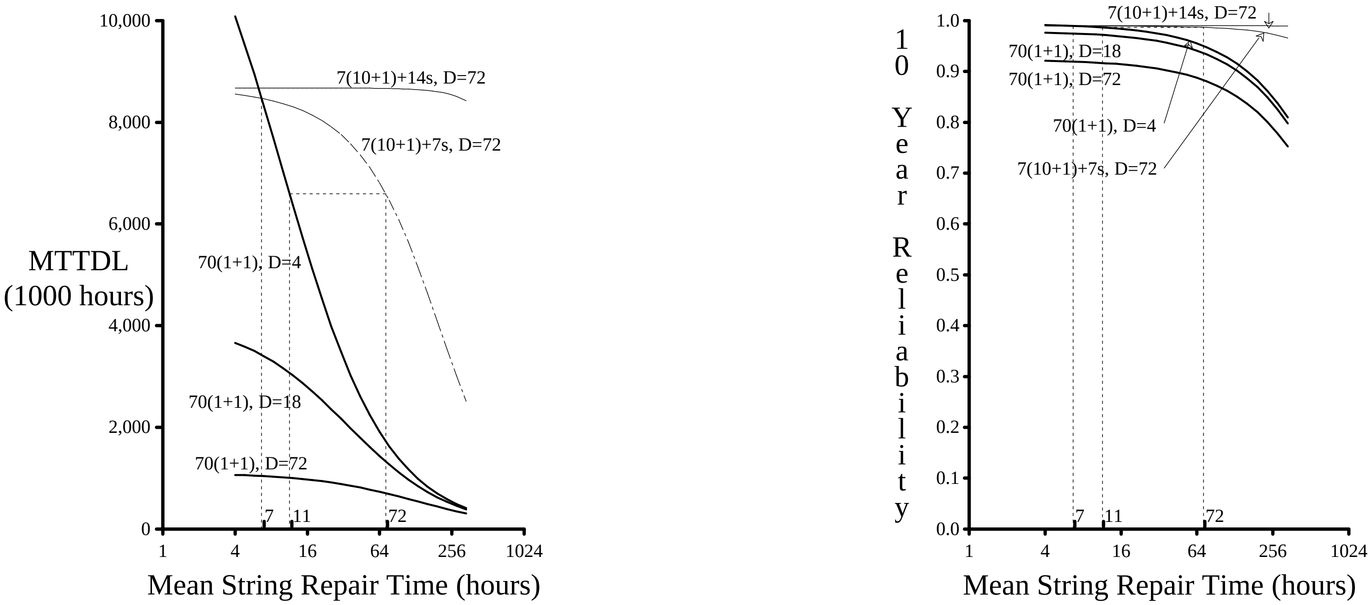


Figure 17: MTTDL with One String of Spares. *This figure shows the relative difference between the estimate for mean lifetime with one string of spare disks given in Equation 23 and the estimate for mean lifetime with an infinite number of strings of spare disks given in Equation 22. Both figures employ the same 100 parameter sets selected at random from a large collection of conceivable parameter sets. One parameter set is not included in this figure because its relative difference is 520%. This parameter set has unusually frequent and long string repairs. Its values are: three parity groups of 21 disks each, an $MTTF_{disk}$ of 100,000 hours, an $MTTF_{string}$ of 50,000 hours, an $MTTR_{disk-recovery}$ of 0.5 hours, an $MTTR_{string}$ of 168 hours, and a replacement-disk delivery time of 8 hours.*



Figures 18a and 18b: N+1 Parity versus Mirrored Reliability. *Figure 18a, on the left, and Figure 18b, on the right, show the effect of string repair time and replacement-disk delivery time on mean lifetime and reliability, respectively, in N+1-parity and mirrored disk arrays. Our strawman disk array contains seven parity groups each with 10 data disks and a parity disk (7(10+1)). Each disk and each string has an exponentially distributed lifetime with mean 150,000 hours and disk recovery time is exponentially distributed with mean one hour when a spare disk is available. The figures show two examples of our strawman disk array: one with one string of spare disks (+7s) and the other with two strings of spare disks (+14s). In both cases, replacement-disk delivery takes 72 hours (D=72). A comparable mirrored disk array has 70 parity groups each with one data and one duplicate disk (70(1+1)). Because each string contains seven disks, the mirrored disk array has 20 strings. These figures show three examples of a comparable mirrored disk array with replacement-disk delivery times of (D=) 4, 18, and 72 hours. In all arrays string-repair time is exponentially distributed with its mean displayed on the x-axis.*

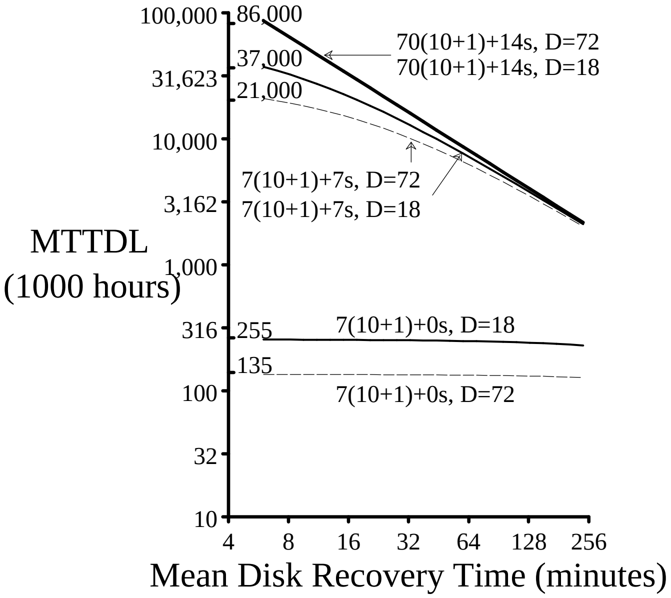


Figure 19: Disk Recovery Time versus On-line Spares. *This figure shows the effect on the mean lifetime of an N+1-parity disk array of faster disk recovery. Our strawman disk array usually assumes an average disk recovery time of one hour. In this figure the disk recovery time is varied from about six minutes to four hours. It presents seven variations on our strawman disk array: arrays with zero, one, and two strings of spare disks are shown with replacement-disk delivery times of 18 hours and 72 hours*

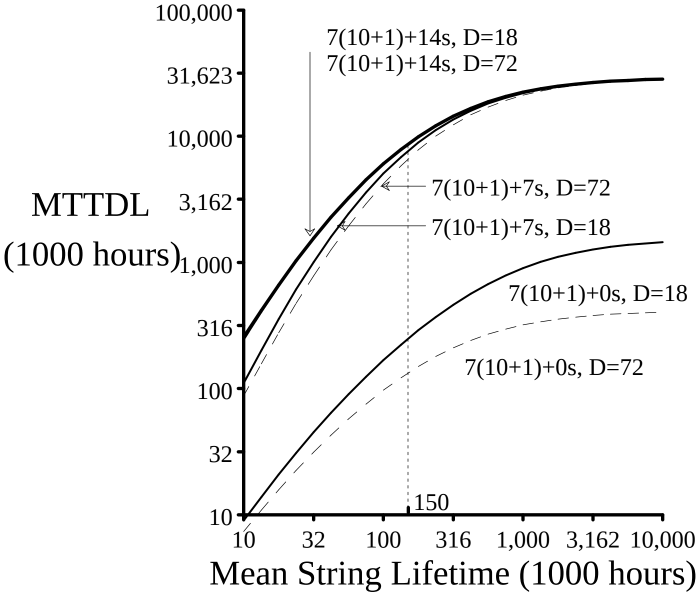
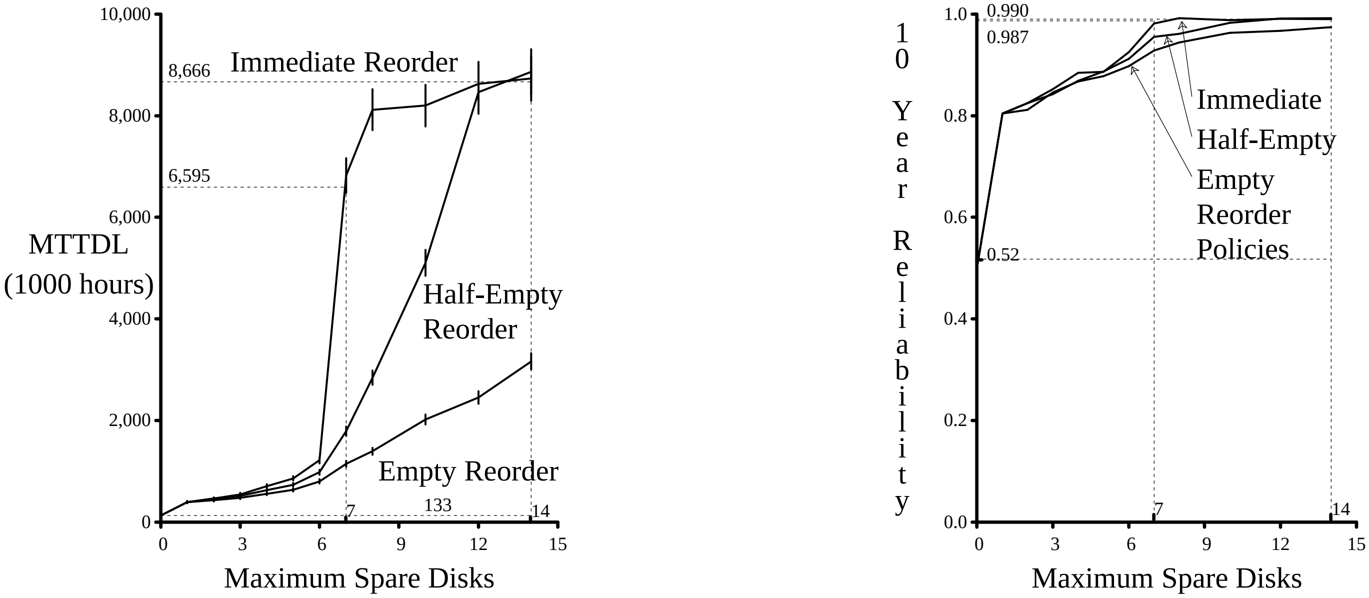


Figure 20: Diminishing Effect of More Reliable Strings on Array Reliability. *This figure shows the effect of string reliability on the mean lifetime of our strawman disk array. Seven variations for the disk array are shown. These are the same configurations presented in Figure 19 of the last section. In this figure, however, mean disk-recovery time is again fixed at one hour and mean string lifetime is varied instead. The dotted vertical line shows the default mean string reliability in our strawman disk array.*



Figures 21a and 21b: Partially Populated Spare Strings and Low Reorder Thresholds. *This figure shows simulated mean lifetime, on the left in Figure 21a, and 10-year reliability, on the right in Figure 21b, for our strawman disk array as a function of the maximum number of spare disks. When the maximum number of spare disks is less than 7 or between 7 and 14, there is a string partially populated with spare disks. Three variations are presented based on the reorder threshold: first, a threshold of one less than the maximum number of spare disks (immediate reorder), second, a threshold that is the integer part of half of the maximum number of spare disks (half-empty reorder), and third, a threshold of zero (empty reorder). Each estimated MTTDL generated by simulation is marked by a vertical bar showing the 95% confidence interval. Note that these curves may not be strictly monotonic increasing because of this variance in simulated estimates. Dotted lines show modelling estimates for zero, one, and two strings of spare disks.*

MODEL	MTTDL	RELIABILITY			OVER-
	(hours)	1 year	3 year	10 year	HEAD
No Redundancy					
One Disk	150,000	0.94	0.84	0.56	0%
Seventy Disks	2,143	0.02	0.00	0.00	0%
Independent Disk Failures Only, (N+1 = 11, D = 72, $MTTR_{disk-recovery} = 1$)					
0 Spares	411,444	0.98	0.94	0.81	10%
1 Spares, 0 Thresh.	12,734,300	0.9993	0.9979	0.9931	11%
2 Spares, 0 Thresh.	17,568,200	0.9995	0.9985	0.9950	13%
2 Spares, 1 Thresh.	28,758,300	0.9997	0.9990	0.9970	13%
∞ Spares	29,224,900	0.9997	0.9991	0.9970	∞
Independent and Dependent Disk Failures, ($MTTF_{string} = 150,000$, $MTTR_{string} = 72$)					
0 Spares	133,235	0.94	0.82	0.52	10%
7 Spares, 6 Thresh.	6,594,890	0.999	0.996	0.987	20%
14 Spares, 13 Thresh.	8,665,860	0.999	0.997	0.990	30%
∞ Spares	8,673,790	0.999	0.997	0.990	∞

Table 2: Summary of Reliability Estimates for Strawman Disk Array. *This figure summarizes mean time until data is lost and reliability estimates from each of the models in this paper applied to our strawman disk array first presented in Table 1 of the introduction to this paper. The last column shows the overhead cost of redundancy as a percentage of the non-redundant disk array cost. In this table ‘Spares’ is the maximum number of spares and ‘Thresh.’ is the reorder threshold. This disk array has 70 data disks organized into an orthogonal array of seven parity groups with 10 data disks and a parity disk in each group. Disks have exponentially distributed lifetimes with a mean of 150,000 hours. Strings have exponentially distributed lifetimes with the same mean. Disk recoveries and string repairs have exponentially distributed durations with means one hour and 72 hours, respectively. Replacement-disk delivery time is the minimum of a fixed 72 hour period,* D*, or the time until an already issued order arrives.*