Data integrity analysis of disk array systems
with analytic modeling of coverage

Manish Malhotra a,*, Kishor S. Trivedi b

a Department of Computer Science, Duke University, Durham, NC 27706, USA
b Department of Electrical Engineering, Duke University, Durham, NC 27706, USA
Data integrity analysis of disk array systems with analytic modeling of coverage

Manish Malhotra a,*, Kishor S. Trivedi b

a Department of Computer Science, Duke University, Durham, NC 27706, USA
b Department of Electrical Engineering, Duke University, Durham, NC 27706, USA

Abstract

Detailed dependability models of various disk array organizations are developed taking into account both the hard disk failures and transient errors. Various error and failure modes of individual disks and the disk array are identified. A small proportion of transient errors account for uncovered failures in a disk array. This leads to analytic computation of probability of disk failures based on several factors including byte error rate of a disk and ECC code etc. Traditionally used measures like MTTDL and data availability remain virtually unchanged with change in mean recovery time. New dependability measures such as degraded capacity time are considered to bring out the effect of mean recovery time on dependability of disk arrays. Our analysis also reveals that mirrored disk organization has higher MTTDL than other disk array organizations if the only failure mode considered is data loss while catastrophic errors are ignored. However, if catastrophic errors are taken into account, then RAID-3,4,5 organizations have higher data-integrity than other disk array schemes. We also develop models that take into account the reliability of support hardware components and different placement schemes for arranging support hardware such as power supply. Our analysis reveals that RAID-1 benefits the most from orthogonal placement of support hardware.

Keywords: Availability; Data integrity; Dependability; Disk arrays; RAID; Reliability

1. Introduction

Various fault-tolerant high performance disk array architectures have been proposed to bridge the gap between high CPU performance and low I/O performance and at the same time maintaining high data reliability and availability as conventional disks [19]. Some disk array architectures have been proposed in [6,13,15,18,19,23]. Patterson et al. [19] coined the term RAID (Redundant Array of Inexpensive Disks) for fault-tolerant disk array systems with redundancy. They unified different disk system architectures as different levels of RAID (levels 1, 2, 3, 4 and 5).
Some studies have been done in the past to analyze disk array reliability. Gibson [3] and Patterson et al. [19] have analyzed reliability of RAID in terms of mean time to data loss (MTTDL). Bitton and Gray [2] have analyzed MTTF for mirrored disks (RAID-1). Gibson and Patterson [5] provide a comprehensive analysis of RAID-1 and RAID-5 reliability and show that RAID-5 organization with a few spares yields higher reliability than RAID-1 at lower cost. However, their results are based on a simple approximation. Assuming that the time to failure of a group of disks is exponentially distributed, they compute the reliability of disk arrays using the approximate value of MTTDL. Usefulness of these approximations is that MTTDL and reliability are expressible in closed form. However, even when the individual disk failure times are exponentially distributed, the time to failure of a group of disks is in general not exponentially distributed.

Ng [17] analyzed the effect of sparing (number of spares and whether spares are shared among groups or not) on reliability of disk arrays. He concluded that nearly all the improvement in disk array reliability is achieved by adding one spare. We reached similar conclusion with a more detailed model [14]. Another result was that making individual disks more reliable improves disk array reliability only to a certain extent. Schulze [24] showed that reliability of support hardware components such as the power supply and cooling equipment could severely degrade the reliability of disk array. They proposed an organization scheme in which support hardware components are placed orthogonal to parity groups of disks. We develop reliability models for disk arrays that take into account failure, repair, and placement schemes of support hardware components. An approximate model for reliability of disk arrays with orthogonal placement of support hardware is developed.

Traditionally, only hard disk failures have been considered to analyze the reliability of disk arrays. However, it is the transient faults that could lead to a catastrophic error such as incorrect data being passed to the user. We develop models which consider both the transient errors and hard failures. Modeling of various failure modes leads to better understanding of failure and recovery behavior of disk arrays and computation of dependability measures such as data-integrity and mean time to catastrophic error besides the commonly used dependability measures such as mean time to data loss and data-reliability. It was shown in [14] that reliability of disk arrays can be greatly improved with improved coverage. However, the analysis assumed constant coverage. In this paper, we analytically model coverage of disk errors based on how transient errors are detected and corrected. Analytic modeling of coverage reveals the parameters on which the coverage depends and analysis of dependability models properly identifies the bottleneck parameters.

One of the conclusions in [14] is that reliability of disk arrays does not improve significantly with the decrease in mean recovery time from disk failures. This leads to the conclusion that it is not worth taking expensive measures to reduce the recovery time from the errors as far as data reliability and availability are concerned. On the other hand, it has been shown that performance of disk array degrades significantly if a single disk in a group fails. New designs of disk arrays to improve performance of a disk array in degraded mode have been proposed [21]. This suggests that the time spent in a degraded state if it is reached should be as small as possible. To incorporate this in our analysis, we consider degraded capacity time as a dependability measure of interest. This measure brings out the effect of data reconstruction time and illustrates the need for a proper choice of metric for comparison and evaluation.
We develop hierarchical reliability models for a general disk array architecture as described in [11] and specialize these models to specific disk array organizations (classified as various levels of RAID). In Section 2, we describe a hierarchical reliability model for a general disk array architecture. In Section 3, we develop reliability models for different RAID architectures with finite number of spares. This model can be specialized to the case of cold or hot spares. In Section 4, numerical results are presented and discussed. Conclusions are presented in Section 5.

2. Disk errors and failures

Errors in the data read from a disk could arise from two factors: either the correct data is read incorrectly or data was originally stored incorrectly (i.e., data was incorrectly written onto a disk). Some of the major factors that contribute to errors in the data read from or written to disks are:

- Imperfections on disk surface due to poor magnetic coating. Defects on disk surface could be caused either due to manufacturing defects or over a period of time due to wear out or both.
- Weak heads are more susceptible to electronic noise and peak-shift error [10]. The heads could be weak from the start due to manufacturing defects and also wear out over a long period of operation.
- Electronic noise causes data bits to be read or written incorrectly. The larger the noise, the more are the chances of bit errors.
- Difficult data patterns increase the chances of errors in data. Howell [10] artificially created some such data patterns to (successfully) cause data errors in his measurement study.
- Improper horizontal alignment of head with the track during the seek operation on the disk causes errors. Typically, this occurs relatively frequently, but does not persist, i.e., a re-seek would cause the head to align properly with the appropriate track. If misalignment of a head persists, then the head is considered failed and must be replaced.
- Abnormal vertical elevation of head from the disk surface would cause errors. If the elevation of the head is higher than desired, then the read or write performed is relatively weaker [1]. If this problem of improper elevation persists, then the head must be replaced.

Usually, the severity of any of the above mentioned factors may not be enough to cause data errors. However, errors are mostly caused when two or more of these factors have a combined effect. Difficult data patterns [10] increase the chances of a weak head reading or writing erroneous data. Data written by a head at an abnormally high elevation is likely to be read incorrectly in presence of large electronic noise. Similarly, a weak head reading data from an imperfect disk surface is likely to read incorrectly.

An error may be transient or permanent and sometimes the fault may be so severe that it necessitates disk replacement. We now classify various disk errors and failures and the recovery actions required in each case.

2.1. Failure modes of individual disks

For the sake of illustration, we assume an interleaved Reed–Solomon code (cf. [20, pp. 327]). This is a single error correcting and double error detecting code. It also corrects any error burst
confined to three consecutive bytes and detects any error burst confined to six consecutive bytes. A random-byte error could be a $k$-byte error ($k = 0, 1, 2, \ldots, n$ where $n$ is the number of bytes in a block including ECC bytes). The probability of any pattern of $k$-byte error occurring is

$$p_k = \binom{n}{k} p_b^k (1 - p_b)^{n-k}, \quad k = 0, 1, \ldots, n,$$

(1)

where $p_b$ is the probability of error in a single byte.

The proportion of various kinds of transient errors can be computed by conditioning on the event that an error has occurred:

$$P(\text{at least one error}) = 1 - p_0 = 1 - (1 - p_b)^n.$$  

(2)

The proportion of transient errors that are $k$-byte errors is then given by

$$q_k = \frac{p_k}{1 - p_0} = \frac{\binom{n}{k} p_b^k (1 - p_b)^{n-k}}{1 - (1 - p_b)^n}.$$  

(3)

Various errors and failure modes can be classified as

- Transient detectable and correctable errors. All the single byte errors are detected and corrected. These errors cause no damage.

- Transient detectable but uncorrectable errors. Double-byte errors are detected but the decoder knows that these errors cannot be corrected by the code by itself. It is likely that such an error is caused by improper alignment of the head with the track or some such temporary condition which would hopefully be corrected if a reread is carried out. The probability of occurrence of a double-byte error is $q_2$. A transient uncorrectable error would be either corrected by retry attempts or would be declared permanent. If $p_{tc}$ is the proportion of errors which will be corrected upon a retry (retry correctable errors), then probability of transient uncorrectable errors which are corrected upon a retry is

$$p_1 = p_{tc} q_2 = p_{tc} \frac{\binom{n}{2} p_b^2 (1 - p_b)^{n-2}}{1 - (1 - p_b)^n}.$$  

(4)

- Non-transient detectable but uncorrectable errors. A small percentage of uncorrectable errors may not be corrected despite several attempts to reread the block of data (e.g., if the head is permanently damaged and misaligns each time or if the sector is permanently damaged). Such an error is termed permanent (or non-transient). The probability of a permanent uncorrectable error is

$$p_0 = (1 - p_{tc}) q_2 = (1 - p_{tc}) \frac{\binom{n}{2} p_b^2 (1 - p_b)^{n-2}}{1 - (1 - p_b)^n}.$$  

(5)

These errors are usually corrected by recovery actions such as constructing the data stored on a bad sector (track) on an empty sector (track) on the same disk by using the data (and
parity information if any) stored on other operational disks in the disk array or by replacing the head if necessary. Recovery from permanent errors takes longer since it involves copying a bad sector (or a track or more) onto some other part of the disk. Mean recovery time from permanent errors is larger than the mean recovery time from transient retry correctable errors. We consider the disk failed if such an error occurs. However, the recovery is fast since disk replacement is not necessary.

- Detectable but miscorrected errors. Random three-byte errors are miscorrected by the code if nearest-neighbor decoding is employed [26]. This results in a catastrophic error since incorrect data is passed to the user. The probability of a miscorrected error is

\[
q_3 = \frac{{n \choose 3} p_b^3 (1 - p_b)^{n-3}}{1 - (1 - p_b)^n}.
\]  

(6)

Recovery action in this case is initiated only upon some external input such as a user complaint. It may not be possible to track the error in this case and a safe recovery procedure would be to check (and replace if necessary) the ECC circuitry including the decoder and reconstructing the data onto a spare disk from the backup tape. Recovery time from such errors could be quite high because of the exceptionally hazardous nature of such errors.

- Undetectable errors. Four-byte errors could change a code word to another code word in which case the error goes undetected. These errors are also catastrophic errors since incorrect data is passed to the user. The probability of an undetected error is

\[
q_4 = \frac{{n \choose 4} p_b^4 (1 - p_b)^{n-4}}{1 - (1 - p_b)^n}.
\]  

(7)

As discussed earlier, ECC codes may not be powerful enough to detect certain malevolent patterns of byte errors. Failure of ECC circuitry is mostly detectable and decoders could be self-checking. However, an undetected failure of ECC circuitry could also cause an undetected or a miscorrected error. Recovery action in this case could be the same as in the case of miscorrected errors. The probability of catastrophic errors is very low and it can be made arbitrarily small by introducing more redundancy (and longer ECC codes with enhanced error-detection capabilities) [1]. Five or more random byte errors are even rarer. Some of these errors will be detected, but for sake of conservativeness, we assume that all three or more byte errors cause a catastrophic error. The probability of occurrence of three or more random byte errors is given by

\[
p_{ce} = \sum_{k=3}^{n} \frac{{n \choose k} p_b^k (1 - p_b)^{n-k}}{1 - (1 - p_b)^n} = (1 - q_1 - q_2).
\]  

(8)

- Burst errors. So far we have considered only random byte errors. Probability of occurrence of multiple random errors is quite low. However, it is the burst errors [20] that dominate the errors occurring in the disks. It is easy to see that if the code does not detect such errors,
then the probability of catastrophic errors could be quite high. Many codes have been designed to detect and correct burst errors of long lengths. In particular, the Reed–Solomon code that we consider is capable of correcting any error burst confined to three consecutive bytes and detecting any error burst confined to six consecutive bytes.

Let $r_k$ be the probability that a burst error is of length $k$ consecutive bits. Any error burst up to 17 bits cannot span more than 3 consecutive bytes. However, a burst error of 18 consecutive bits will span 4 consecutive bytes if it begins at the last bit of a byte. Assuming that the probability of a burst error starting at any of the 8 bit positions in a byte is equal, the probability that an 18 bit burst error will span 4 consecutive bytes is given by $\frac{1}{8}$. Similarly, the probability that a 19 bit burst error will span 4 consecutive bytes is $\frac{2}{8}$. Any burst error larger than 25 bits and less than 41 bits will always cause a detectable but uncorrectable error. Proceeding in this fashion, the probability of a detectable but uncorrectable error (i.e., an error that spans at least 4 and at most 6 consecutive bytes) is given by

$$p_{dx} = \frac{r_{18}}{8} + \frac{r_{19}}{4} + \frac{3r_{20}}{8} + \frac{r_{21}}{2} + \frac{5r_{22}}{8} + \frac{3r_{23}}{4} + \frac{7r_{24}}{8} + \sum_{i=25}^{41} r_i.$$  \hfill (9)

Similarly the probability that a burst error is undetected can be calculated. It is given by

$$p_{ux} = \frac{r_{42}}{8} + \frac{r_{43}}{4} + \frac{3r_{44}}{8} + \frac{r_{45}}{2} + \frac{5r_{46}}{8} + \frac{3r_{47}}{4} + \frac{7r_{48}}{8} + \sum_{i=49}^{K} r_i,$$  \hfill (10)

where $K$ is the number of consecutive bits in a data block. The correctable burst errors may be miscorrected. However, the probability of this event is extremely small. Hence in our model, the undetected and miscorrected random byte errors and undetected burst errors result in a catastrophic error.

- Hard failure of the entire disk, e.g., a head crash on disk surface or data corruption on the entire disk due to failure of power supply or cooling equipment. We assume that these failures are always detected when they occur. The only recovery action possible in this case is replacement of the failed disk and reconstructing data on the spare disk. It is reasonable to assume that all the hard failures are perfectly covered (i.e., detected) since the disk controller should be able to detect if a hard failure occurs.

2.2. Disk arrays

Fault-tolerant disk array systems possess external redundancy either by replicating the data or by storing a parity block for several data blocks which can be used to correct single block errors and detect double block errors. Additional fault-tolerance can be provided using more redundancy [4]. We consider disk arrays that provide single correction and double detection since these are the most commonly used. A disk array is considered failed if data loss occurs or data is unavailable. If a disk in a group has failed (due to a hard failure) or if a permanent error has occurred in the disk, then data stored on this disk is inaccessible and the disk is considered failed. However, the data stored on this disk can be reconstructed using the data on other disks in the group and the parity. Thus, the I/O requests to this disk are satisfied by data reconstruction mechanism.
Meanwhile appropriate recovery action for the failed disk is initiated. If another disk in the group fails (a hard failure or a permanent error occurs) before this recovery action is completed, then data can not be reconstructed. In this case, the disk array is considered failed. Recovery action in this case consists of retrieving whatever data can be retrieved from the backup storage and copying it onto spare disks if disk replacement is necessary. A catastrophic error in any disk in a group constitutes a catastrophic error in the entire disk array. Recovery action in this case is not initiated until the error is detected by some external source (such as the user).

Besides disks, other components in the I/O subsystem may also fail. These include disk controller, actuator, power supply, and cooling equipment among several others. Failure of any of these components constitutes a single point of failure in the I/O subsystem unless redundancy is used. Later in this paper, we present reliability models that include failures of support hardware components.

3. Dependability analysis of disk arrays

3.1. Dependability measures of disk arrays

The dependability of disk arrays can be analyzed in terms of different measures. Data-loss reliability \( L(t) \) at time \( t \) is defined as the probability that no data loss has occurred until time \( t \). Correspondingly, we have mean time to data loss (MTTDL). Data-integrity \( I(t) \) at time \( t \) is defined as the probability that no incorrect data has been passed to the user until time \( t \). Any catastrophic error causes the disk array to lose its data integrity. Correspondingly, we have mean time to catastrophic error (MTCE). Data-availability \( A(t) \) at time \( t \) is defined as the probability that data is accessible at time \( t \). This is a measure of instantaneous availability of data at time \( t \). Interval data-availability \( IA(t) \) is defined as the proportion of the time the data is accessible during interval \( (0, t] \). This is computed by taking the time average of instantaneous data availability over the interval \( (0, t] \). Degraded-capacity time [7] is the annual amount of time during which the disk array is functioning but operating at less than full capacity. It is computed by summing the steady-state probabilities of the disk array being in degraded (but not failed) states and multiplying it by the number of seconds in a year.

3.2. Dependability models of disk arrays

We construct general dependability models which can be specialized to various fault-tolerant disk array organizations. A disk array is organized as \( N \) groups and each group consists of \( G \) disks including the data and check disks. Let \( \lambda_r \) be the rate of occurrence of random byte errors in a block of data. If a block contains \( n \) bytes of data and \( \lambda_b \) is the rate of occurrence of a byte error, then \( \lambda_r = n \lambda_b \). Rate of occurrence of transient detectable but uncorrectable errors is

\[
\lambda_t = p_r \lambda_r.
\]  

(11)

Rate of occurrence of non-transient detectable but uncorrectable errors is \( p_n \lambda_r \). Let \( \lambda_s \) be the
rate at which burst errors occur in a data block. The rate of detectable and uncorrectable burst errors is \( p_{ds} \lambda_x \). The rate of undetected burst errors is \( p_{ux} \lambda_x \). Assume that the burst errors are non-transient and that the recovery time from non-transient random byte errors and burst errors is nearly the same. Further assuming that burst errors occur independently of random byte errors, the rate of detectable and uncorrectable errors in a disk is

\[
\lambda_p = p_p \lambda_r + p_{ds} \lambda_x.
\]

(12)

Similarly, the rate of catastrophic errors is

\[
\lambda_{cf} = p_{ce} \lambda_r + p_{ux} \lambda_x.
\]

(13)

Let \( \lambda_b \) be the rate of occurrence of hard disk failures.

Assuming that each group of disks is statistically independent of other groups, the overall reliability of disk array is modeled by a simple reliability block diagram (RBD) shown in Fig. 1. Each block of this RBD represents a group of disks. Assume that groups behave independently of each other. If \( R_i(t) \) is the reliability of group \( i \), then reliability of the disk array is given by

\[
R_{da}(t) = \prod_{i=1}^{N} R_i(t).
\]

(14)

The reliability of a group of disks is obtained by solving the Markov models described next. The mean time to data loss for the disk array can be computed as:

\[
MTTDL_{da} = \int_{0}^{\infty} R_{da}(t) \, dt.
\]

(15)

3.2.1. Unlimited number of spares or no spares

The dependability model for a group of \( G \) disks is shown in Fig. 2. This model is valid under two different assumptions regarding spare disks: (1) No spare is maintained and each time a disk fails, a spare is ordered from the manufacturer. The recovery time in this case is quite high. (2) Unlimited number of cold spares is maintained. In practice, this scenario is well-approximated by maintaining a minimum number of cold spares. Each time a disk fails, a spare is used up and a new spare is also ordered from the manufacturer. The probability of several disks failing at the same time is extremely low. Hence, this strategy ensures that almost always, a spare will be available when a disk fails. By increasing the minimum number of cold spares that must be maintained, the probability that no spare is available when it is needed can be made arbitrarily small. The recovery time is much less if cold spares are available and do not have to be ordered from the manufacturer.
Actually, we are making an approximation in this model for the recovery action. If no spares are maintained, then recovery action consists of three stages: (1) Order the spare from the manufacturer and call the service person, (2) Installation of the spare by a service person, and (3) Data reconstruction on the new disk. We lump all these stages together and consider a single transition for recovery action. This approximation greatly simplifies the model. Comparison of the results show that approximate model yields almost as good results as the exact model. This happens because failure rates of a disk are extremely small compared to recovery rates. Hence, the time spent in degraded states is very small compared to the time spent in the fully operational state.

We assume that an ECC code with single-byte error correction, double-byte error detection and burst error correction is used. We also assume that the probability of an undetected burst error is negligible and that the catastrophic errors result from three or more random multiple-byte errors.

In this model, all the $G$ disks are operational in state 2. In some implementations (such as DEC HSC-70 disk controller), the disk controller has the ability to predict a hard disk failure before it actually occurs [11]. Some implementations utilize this property to prevent data loss and reduce the reconstruction time. Assume that no loss of data occurs if the disk controller correctly predicts an impending hard failure of disk. Further assume that the spare is electronically switched in and data is copied onto the spare before the failing disk is powered.
down. This sequence of operations does not result in a change of state of the system. However, the disk controller may not always be able to predict a disk failure. With probability \((1 - \alpha)\), an impending hard failure is not predicted. An unpredicted hard disk failure of any disk causes a transition to state 3 with rate \(G\lambda_h(1 - \alpha)\) where \(\lambda_h\) is the rate of hard disk failures. If the disk controller has no disk failure prediction capability, then \(\alpha = 0\).

Recovery action is initiated after a disk failure and mean recovery time is \(1/\mu_4\). Rate \(\mu_4\) depends upon the availability of spares, type of spares (hot or cold), and data reconstruction time. In reality, the recovery time distribution is closer to deterministic than exponential. However, it has been shown through simulation [5] that using exponential recovery time yields results that are almost as accurate as using constant-time recovery. In this model, we assume that a detected transient single-byte error which is correctable by the ECC code does not result in a change of state of the system. This is quite reasonable since the recovery time in this case is negligible. Double-byte errors (detected but not corrected) cause a transition to state T2 with rate \(G\lambda_t\). Recovery from state T2 is accomplished by a retry and mean time to complete a retry is \(1/\mu_r\). Three or more random byte errors cause transition from state 2 to state CF with rate \(G\lambda_{\text{ef}}\). A detected but uncorrectable permanent (random byte or burst) error in state 2 causes transition to state 4 with rate \(G\lambda_p\). Rate of recovery from such a fault is \(\mu_5\). It is easy to see that recovery rates from different failure states are different, specifically we have \(\mu_4 < \mu_5 < \mu_r\).

In state 3, an unpredicted hard disk failure with rate \((G - 1)\lambda_h(1 - \alpha)\) would cause a transition to the data loss state labeled 7. Data is lost since two disks in the group have failed. Recovery action in this case necessitates accessing the backup tape to recover the lost data if possible. Both the failed disks are replaced and data is reconstructed. Recovery causes transition back to state 2 with rate \(\mu_2\). A detected but uncorrectable permanent (random byte or burst) error in state 3 with rate \((G - 1)\lambda_p\) causes transition to another data loss state 6. In this data loss state, there are two disks in failed state. The disks with hard failure must be replaced, but the other disk requires a simpler recovery action such as copying a bad sector to some empty sector on the disk. Thus, recovery rate from state 6 is \(\mu_1\) where \(\mu_1 > \mu_2\). A catastrophic error in state 3 or 4 with rate \((G - 1)\lambda_{\text{ef}}\) causes transition to state CF.

In state 4, a detected but uncorrectable permanent (random byte or burst) error with rate \((G - 1)\lambda_p\) causes transition to data loss state 5. An unpredicted hard disk failure with rate \((G - 1)\lambda_h(1 - \alpha)\) causes transition to data loss state 6. Recovery rate from state 5 is \(\mu_3\) where \(\mu_2 < \mu_1 < \mu_3\). In states 3 and 4, detected but uncorrectable transient errors occur at rate \((G - 1)\lambda_t\) and cause transitions to states T3 and T4 respectively.

The probability of being in state CF at time \(t\) gives complement of data-integrity of a group at time \(t\). To compute instantaneous data-availability at time \(t\), we sum the probabilities of being in states 2, 3, and 4 at time \(t\). To compute the degraded capacity time, we make the state CF non-absorbing, i.e., we allow recovery from this state to state 2. From this availability model, we compute the sum of steady-state probabilities of being in states 3 and 4, and multiply it by the number of seconds in one year. This yields the number of seconds the disk array is functioning at degraded capacity in one year. To compute data loss reliability at time \(t\), we make the data loss states 5, 6, and 7 absorbing and merge these states and CF state into a single absorbing state. This is accomplished by deleting the outgoing arcs (recovery) from these states. Probability of being in the absorbing state at time \(t\) yields data-loss reliability. Here we consider catastrophic errors as part of data loss.
3.2.2. Finite number of spares

If we consider finite number of spares per group, then we obtain the dependability model shown in Fig. 3. Each state is labeled as a two-tuple \((i, j)\) where \(i\) is the number of data and check disks in a group and \(j\) is the number of remaining spares in the group. A state \((i', j)\) indicates that one disk in the group is undergoing recovery from a permanent error. A state \((i'', j)\) indicates that two disks in a group are undergoing recovery from permanent errors. \(M\) is the number of spares in the beginning. All the states labeled \((i - 2, j)\) (two hard disk failures), \((i' - 1, j)\) (one hard disk failure and one permanent error), and \((i'', j)\) (two permanent errors)

![Diagram](image)

Fig. 3. Dependability model for a group of disks with finite number of spares.
are data loss states. Once again, we are being conservative in assuming that all states with two permanent errors result in data loss. By data loss states, we mean that data cannot be reconstructed. The states represented by circles are transient error states from which recovery is successful upon retry.

If the disk controller has the ability to predict hard disk failures beforehand, then there also exists the possibility of erroneous prediction, i.e., a disk is predicted to fail even when it is not going to. In infinite spares model (Fig. 2), this does not affect the model (although it affects the cost). However, in finite spares model, erroneous prediction will result in consumption of spares and must be appropriately reflected by state change. Rate \( \lambda_{sp} = i \lambda_{sp} + G \lambda_{s} \alpha + \gamma \), where \( 1/\gamma \) is the mean time to next erroneous prediction.

If hot spares are used, then \( \lambda_{sp} \) denotes the failure rate of spare disks. Hot spare disks can fail only in the hard failure mode. We take \( \lambda_{sp} < \lambda_{h} \), i.e., spares fail at a lower rate than active disks. If cold spares are used, then \( \lambda_{sp} = 0 \). Hot spares can be electronically switched in when needed and the recovery will consist of simply data reconstruction. The rates \( \mu_{2}, \mu_{1}, \) and \( \mu_{4} \) are much lower for hot spares than for the cold spares.

The assumption here is that spares are not replenished after consumption. If a disk fails after all the spares are exhausted, then the spares must be ordered from the manufacturer. Therefore, the recovery rates \( \mu_{2} < \mu_{1} \) and \( \mu_{3} < \mu_{2} \). We have also assumed that the spares are not used to store any data. Hence the failure of any spare disk does not cause transition to a degraded states. All the states labeled \( (G, M), (G, M - 1), \ldots, (G, 0) \) are fully operational (non-degraded) states of the disk array. Some of the sparing schemes such as distributed sparing [16] use spare disks to store data as well. In those cases, failures of spares disks will cause the transition to degraded states. Spares in such schemes can also have permanent error and transient error modes besides hard failure mode.

These general dependability models of a disk array can be specialized to various disk array organizations classified as different levels of RAID by Patterson et al. [19]. We briefly describe various disk array organizations. For all the different organizations, we assume that during data reconstruction after a disk in a group has failed, the data is kept available, i.e., user requests are also served along with recovery requests although at a low performance level. For this case, the data reconstruction time for different disk array organizations is different. However, if two disks in a group fail, then data becomes unavailable. Disk array does not serve user requests until it becomes fully operational again. For this case, we assume that mean data reconstruction time for different disk array organizations is the same, i.e., rates \( \mu_{3}, \mu_{2} \) and \( \mu_{3} \) are equal for RAID-1,2,3,4, and 5.

4. Disk array organizations

4.1. RAID-1 (duplexed data)

Mirroring (or duplexing) is the simplest although an expensive approach to improve the reliability of disk systems [2,12]. For each data disk, another disk that contains an identical copy of the data is maintained. According to RAID configuration, each pair of mirrored disks forms a group. There is no interleaving of data. If one of the disks in a group fails, then a spare is switched in. The data reconstruction consists of copying the data onto the spare from the other
working disk. The mean recovery time (including the time to switch in the spare and copy the data) for RAID-1 after a hard disk failure has occurred is taken to be one hour and 10 minutes (i.e., $\mu_4 = 0.8571$ per hour) assuming that a spare is available on-line. The rough estimate is that it takes about an hour to switch in the spare and about 10 minutes to copy the data onto the spare. Rates $\mu_2 = 0.24$ and $\mu_1 = 0.4615$ per hour. Depending upon the associated hardware like channels and disk controllers, I/O requests to different groups can be processed in parallel. The number of disks in each group is $G = 2$.

4.2. RAID-2 (Hamming coded ECC)

In this organization, each group consists of $D$ data disks and $C$ check disks where $C \geq \log_2(D + C + 1)$. The data stored on check disks can be used to correct single block errors and detect double block errors. Failure of a single disk in a group is tolerable since data on failed disk can be reconstructed. However, failure of more than one disk in a group causes data loss. The number of disks in each group is $G = D + C$. A minor advantage of RAID-2 over RAID-3,4,5 is that transient detectable but uncorrectable errors can be corrected on-the-fly without having to resort to rereads. In the models shown, this implies that $\lambda_1 = \lambda_2 = 0$. Mean time after a hard disk failure in RAID-2 takes longer than any other RAID level because of more check data. We choose this mean recovery time to be two hours and 10 minutes (i.e., $\mu_4 = 0.4615$ per hour). Rates $\mu_2 = 0.1935$ and $\mu_1 = 0.3158$ per hour.

4.3. RAID-3 (bit-interleaved data)

The $C$ check disks in RAID-2 are mainly required to detect the failed disk. However, the disk controller can detect the failed disk either by special signals or by the ECC information stored on each sector of the disk to correct transient errors. To correct single block errors, a single parity disk suffices. In RAID-3, each group consists of $D$ data disks and 1 check disk. The data is bit-interleaved across the data disks on a group. The parity bit for the strip of $D$ data bits, with each bit stored on one data disk, is stored on the check disk. The advantage of this organization is the high bandwidth since data transfer can be performed in parallel. This also implies that to read a block of data, all the disks in the group need to be accessed. Failure of any single disk is tolerable since any bit of a failed disk can be reconstructed by the corresponding data bits on the remaining $D - 1$ data disks and the parity bit on the check disk. The number of disks in each group is $G = D + 1$. Park and Balasubramaniam [18] proposed a RAID-3 architecture. For RAID-3, mean recovery time after a hard disk failure is faster than RAID-5 because there is less positioning overhead since heads need not be re-aligned to reconstruct data on a disk [8]. We choose the mean recovery time to be one hour and 20 minutes (very slightly larger than RAID-1) i.e., $\mu_4 = 0.75$ per hour. Rates $\mu_2 = 0.2307$ and $\mu_1 = 0.4286$ per hour.

4.4. RAID-4 (block-interleaved data)

RAID-4 differs from RAID-3 in the manner in which data is interleaved. Data interleaving in RAID-3 takes place at bit level. This improves the bandwidth but it prevents more than one I/O request to be serviced at the same time. For small reads and writes, RAID-3 could be
inefficient. This prompted another design in which data is interleaved at sector level instead of bit level. To access a data block, only one disk needs to be accessed. If the I/O requests are small, then several read requests can be serviced in parallel. However, writes are still limited to one per group since each write requires parity to be read and new parity (because of new data) to be written. Since parity information is stored on a single disk, writes are not parallelized. The number of disks in each group is $G = D + 1$. Salem and Garcia-Molina [23] proposed a RAID-4 organization. In this case, the mean recovery time depends on how fast the parity information is read from the parity disk. Since there is a single parity disk, the utilization of this parity disk will largely determine $\mu_4$. We choose mean recovery time to be two hours (i.e., $\mu_4 = 0.5$ per hour). Rates $\mu_2 = 0.2$ and $\mu_1 = 0.3333$ per hour.

4.5. RAID-5 (Block-interleaved data and rotated parity)

This organization was proposed by Patterson et al. [19]. RAID-5 differs from RAID-4 in that the parity is also striped across all the disks in a group. This allows writes to be serviced in parallel as well. The data and parity is striped across all the disks in a group. The number of disks in each group is $G = D + 1$. Various strategies have been proposed on how to reconstruct data after a single disk in a group has failed [9]. Depending upon the strategy chosen, the mean data reconstruction time (excluding the time to switch in the spare) would differ. For RAID-5, we choose mean recovery time (including the time to switch in the spare) to be one hour and 30 minutes (i.e., $\mu_4 = 0.6667$ per hour) [9]. Rates $\mu_2 = 0.2222$ and $\mu_1 = 0.4$ per hour. Holland and Gibson [9] present results from a measurement study on data reconstruction times for RAID-5. Therefore, we use data from these results to compute numerical value of $\mu_4$ for RAID-5. Due to lack of measurement data on reconstruction times for other RAID levels, the value of $\mu_4$ used for other RAID levels is computed relative to the value of $\mu_4$ for RAID-5 as pointed out in [8].

5. Disk arrays with support hardware

A disk array system has many hardware components that are needed for proper functioning of the disk array. These include host bus adaptor (HBA), disk array controller (DC), hard disk drive (HDD) controller, single board controller (SBC) (track buffer and error correction circuitry (ECC) are resident in SBC), cooling hardware, and power supply, etc. So far we have considered only the disks in the reliability models of disk arrays. Schulze et al. [24] have shown that failures of the support hardware considerably reduces the overall reliability of a disk array. In fact, failure of some of the support hardware components such as cooling equipment and power supply may result in data loss. In this section, we model failure-repair behavior of support hardware components on data loss reliability of disk array. We consider two hardware organizational schemes.

5.1. Serial placement of support hardware

This is a simple organizational scheme in which the disk array has a set of associated hardware components (HBA, power supply (PS), cooling fans (CF), HDD, SBC, disk controller
Fig. 4. Reliability block diagram of RAID (1, 2, 3, 4, 5) with serial placement of support hardware.

e.g., These components are placed serially with the disk array. If there is no redundancy in these components, then failure of any of these components results in the failure of the disk array. The reliability block diagram of the overall system is shown in Fig. 4. The reliability of the disk array is given by

\[ R_{da}(t) = \left( \prod_{i=1}^{N} R_i(t) \right) R_{hba}(t) R_{dc}(t) R_{sbc}(t) R_{ps}(t) R_{cf}(t). \]  

(16)

We assume that time to failure of each support hardware component is exponentially distributed.

5.2. Orthogonal placement of support hardware

Schulze et al. [24] proposed an organizational scheme for placement of disks and the support hardware in a way that makes disk arrays more fault-tolerant. The disks are organized in a two dimensional grid with each row representing a parity group of disks. In RAID, each parity group can tolerate single disk failures. Support hardware (power supply, cooling fans, HBA, etc.) is provided for each column of disks. Thus, each column forms a support hardware group. This organization is shown in Fig. 5. The orthogonal placement of parity groups against support hardware groups provides fault-tolerance against failure of support hardware components. Disk array is operational even if all the disks in a column group or any support hardware

Fig. 5. RAID organization with orthogonal placement of support hardware.
components along a column fail. However, disk array in this case is more expensive than the serial organization because of large number of associated hardware components.

In [24], an approximate estimate for the MTTF of a disk array organized in this manner is provided. Due to complex dependence of failure and data-reconstruction in this RAID organization, a simple reliability model cannot be developed. We developed a reliability model using stochastic Petri nets but it resulted in a very large Markov chain. The symmetry in this model prompted us to develop a smaller approximate model. The approximate model is obtained by essentially lumping identical states into one state. The approximate Markov model has only six states and yields solutions that were close to the solutions obtained using the exact model. The approximate model for disk array reliability is shown in Fig. 6.

In this model, state UP is the fully operational state of the disk array with no disk or hardware component failed. Assume that all the support hardware components are statistically independent and identical across different columns. Failure of any support hardware component in a column causes the entire column to fail. Assuming exponential time to failure distribution for each of the support hardware component, the failure rate of each hardware column $\lambda_{sh}$ is the sum of failure rate of each component (CF, PS, HBA, DC, etc.) and time to failure distribution of each hardware column is exponential with this rate [25]. In state UP, one of the hardware columns may fail (transition to state S3) and it is repaired at rate $\mu_{sh}$. For the

$$
\begin{align*}
\delta_1 &= G(N - 1)(\lambda_b + \lambda_p) + (G - 1)p_a\lambda_{sh} \\
\delta_2 &= GN\lambda_p \\
\delta_3 &= G\lambda_{sh}(1 - \alpha) \\
\delta_4 &= G\lambda_{sh}p_a \\
\delta_5 &= (G - 1)(p_a\lambda_{sh} + \lambda_b + \lambda_p) \\
\delta_6 &= (G - 1)(p_a\lambda_{sh} + \lambda_b + \lambda_p) \\
\delta_7 &= GN\lambda_{sf} + G\lambda_{sh}(1 - p_a) \\
\delta_8 &= (NG - 1)\lambda_{sf} + (G - 1)(1 - p_a)\lambda_{sh} \\
\delta_9 &= (NG - 1)\lambda_{sf} + (G - 1)(1 - p_a)\lambda_{sh} \\
\delta_{10} &= G(N - 1)\lambda_{sf} + (G - 1)(1 - p_a)\lambda_{sh}
\end{align*}
$$

Fig. 6. Approximate reliability model for orthogonal RAID (1, 2, 3, 4, 5).
lack of real data, we assume that each hardware component has the same MTTR. If this is not the case, then simple extensions to the reliability model can be made by introducing different failure states for failures of different hardware components and their repair.

A failure in a hardware column is covered with probability $p_{oh}$. An uncovered failure causes a catastrophic failure of disk array. While the repair is underway, the disks in this column are considered unoperational. However, if any of the remaining $G(N-1)$ disks fails (i.e., a permanent error or hard failure occurs) or if any of other column hardware groups fails before the repair is completed, then data loss occurs (transition to data loss state DL). In state UP, any of the $G \times N$ disks may suffer a permanent error and transition is made to state S1. Also in state UP, a hard disk failure may occur in any of the $G \times N$ disks. If this hard disk failure is successfully predicted by the controller, then no state change occurs. However, if it is not predicted (with probability $1 - \alpha$), then transition is made to state S2. Recovery from states S1 and S2 occurs at rates $\mu_s$ and $\mu_d$ respectively. If in any of the states S1 or S2, a disk failure (a permanent error or a hard disk failure) occurs among any of the $G-1$ disks belonging to the same group, then data loss occurs and transition to state DL is made. Also, if any hardware component fails in any of $G-1$ columns while the system is in state S1 or S2, then again transition is made to data loss state DL. An uncovered failure of a support hardware component or a catastrophic error in any of the operational disks in states UP, S1, S2, and S3 causes a transition to state CF. In this model, we have ignored transient detectable and retry correctable errors since the time spent in corresponding states is insignificant.

6. Numerical results

We numerically solved the models using the software tool SHARPE [22]. Our aim is to show how disk array reliability depends on various parameters and how important it is to model the effect of these parameters. Comparing with previous studies on disk array studies, we illustrate how these parameters which are not modeled in those studies could change the predictions on disk array reliability. Hence the emphasis should be on the characteristics of the various plots and not on the absolute values of the dependability measures which are used for comparison. In all but one experiments, we found imperceptible difference between MTTDL, MTCE, $L(t)$, and $I(t)$ of RAID-3, RAID-4, and RAID-5.

The numerical values of various parameters unless otherwise specified are: $\lambda_b = 1/40 000$ per hour, $\lambda_s = 1$ per hour, $\lambda_x = 0.2$ per hour, $n = 521$ (512 data bytes and 9 check bytes), $p_b = 10^{-5}$, $p_{ix} = 0.9$, $\alpha = 0.9$, $p_{ox} = 10^{-3}$, $p_{ux} = 10^{-6}$, $\mu_s = 180 000$ per hour, $\mu_x = 1800$ per hour, $\mu_3 = 60$ per hour, $\mu_7 = 1/50$ per hour, and $\mu_8 = 1/72$ per hour. We have chosen a fixed value for the probabilities $p_{ax}$ and $p_{ux}$ rather than compute these values using Eqs. (9) and (10). The purpose of those equations was to illustrate how these probabilities can be computed analytically. However, due to the lack of data from a measurement study, we decided to choose a value directly for $p_{ax}$ and $p_{ux}$ rather than choose values for several different $r_j$'s and then compute $p_{ax}$ and $p_{ux}$.

The mean recovery time after a single hard disk failure is different for different RAID levels. Hence, the values of $\mu_4$, $\mu_1$, and $\mu_2$ are specified in Section 4. Mean recovery time from catastrophic errors is taken to be 18 hours, i.e., $\mu_{ce} = 1/18$ per hour. The base disk array used
for RAID-3,4,5 has $N = 8$ groups of $G = 4$ data disks and one check disk. For RAID-2, $N = 8$ and $G = 7$ (4 data disks and 3 check disks). For RAID-1, $N = 32$ and $G = 2$. The experiments were conducted using the infinite-spares model shown in Fig. 2 unless otherwise specified.

For models with support hardware components, we have from the data provided in [24], MTTF for power supply = 1460 hours, HBA = 123000 hours, power cable = 10000000 hours, SCSI cable = 21000000 hours, cooling equipment = 195000 hours, SBC = 40000 hours and HDD controller = 30000 hours. We take mean repair time for any support hardware component (also MTTR for any support hardware column) to be 24 hours ($\mu_{sh} = 1/24$ per hour).

In the first experiment, we varied the probability of error in a single byte $p_b$. Data-integrity computed at $t = 1000$ hours is plotted as a function of $p_b$ in Fig. 7. We observe that data integrity falls sharply as probability of byte error increases. This means that this is a bottleneck parameter. Probability of byte error can be reduced by periodic maintenance of heads and associated circuitry involved in physically reading and writing data onto a disk. The earlier studies did not model byte errors.

In the next analysis, we considered the effect of mean recovery time on disk array dependability. The mean recovery time considered here is the recovery from a hard disk failure. Recovery from transient errors or permanent errors do not involve disk replacement but recovery from a hard disk failure necessitates disk replacement. Various factors such as as type of spares (hot or cold) and availability of spares greatly influence this recovery time. If no spares are available, then recovery time could be as large as 70 hours or more since the spares are ordered from the manufacturer. On the other hand, if a hot spare is available, then
recovery time could be as low as couple of hours. This recovery time changes the rates $\mu_2$, $\mu_3$ and $\mu_4$. In the Figs. 8 and 9, the x-axis shows the amount of time (in hours) it takes to switch in a spare disk. The time to perform data reconstruction on the spare disk remains the same.

First, we consider data integrity as the dependability measure of interest. Data integrity as a function of mean recovery time is plotted in Fig. 8. This shows that data integrity of disk array organizations remains virtually independent of mean recovery time. RAID-1 has the lowest data integrity compared to various disk array organizations.

Next we consider degraded capacity time as the dependability measure of interest. The plot shown in Fig. 9 reveals a very interesting pattern. Degraded capacity time of disk arrays increases significantly as mean recovery time is increased. This brings out the need to choose an appropriate measure of interest to bring out the effect of certain model parameters. Furthermore, using this measure also brings out the effect of different recovery rates for different RAID levels. RAID-3 has the smallest degraded capacity time among different RAID architectures. Whereas other measures such as data integrity and MTCE show imperceptible change among RAID-3, RAID-4, and RAID-5.

In the next experiment, mean time to catastrophic error (MTCE) of the disk array is analyzed as a function of storage capacity (in terms of number of disk storing data) (Fig. 10). For RAID-2, RAID-3, RAID-4, RAID-5, the number of data disks per group was kept same ($D = 4$) and the number of groups was varied from 4 to 16. For RAID-1, the number of groups was correspondingly varied so that the storage capacity of the disk array is same for various organizations. As
expected, MTCE decreases with increase in total number of disks in a group. This implies that as disk arrays are scaled in size and storage capacity, measures such as more sophisticated ECC schemes [4] or additional redundancy should be taken to ensure high dependability.

Next we analyze mean time to data loss (MTTDL) as a function of byte error rate $\lambda_r$ of the disk. This rate determines the rate of permanent, transient, and catastrophic errors. Transient errors were not modeled in previous studies on disk array reliability. We find that slight increase in this rate causes large decrease in MTTDL.

We perform two analyses. First, we try to recreate the result of earlier studies. We ignore catastrophic errors and consider data loss as the only failure mode of disk array, i.e., delete state CF and merge data loss states into a single absorbing state. MTTDL as a function of $\lambda_r$ is plotted in Fig. 11. RAID-1 has the largest MTTDL among various RAID organizations. This confirms the result of earlier studies [5] that RAID-1 is the most reliable disk array organization.

Now we consider catastrophic errors as well and model those separately. MTCE is plotted as a function of byte error rate in Fig. 12. Interestingly enough, RAID-1 has the smallest MTCE among various RAID organizations. The reason behind this is that disk array organizations do not provide any protection against catastrophic errors. Since RAID-1 has the largest number of groups, the effect of catastrophic errors is most pronounced in RAID-1 compared to RAID-2,3,4,5 (refer to Eq. (14)).

Finally, we analyze the gains in data loss reliability due to orthogonal placement of support hardware components over the serial placement. The disk array reliability is plotted as a
function of time in Fig. 13. Comparing the two curves (e.g., RAID-1 (srl) and RAID-1 (otg)), we find that orthogonal placement of support hardware improves array reliability. A key pattern to note is that for orthogonal organization, RAID-1 architecture has higher reliability than RAID-2,3,4,5 (as opposed to all earlier plots). This happens because there are only two hardware columns in RAID-1. Thus, RAID-1 benefits the most from orthogonal placement as far as improvement in reliability is concerned. Moreover, the overhead cost due to multiple support hardware components (one for each column) of RAID-1 is the least among different RAID organizations. In serial placement of hardware, the reliabilities of different RAID architectures are almost the same. This happens because of the very small MTTF of support hardware (≈ 1460 hours) compared with the MTTF of disks.

7. Conclusions

We have discussed various failure modes and errors in disks. These failure modes are used to analytically model coverage of disk faults. Traditionally, only hard disk failures are considered to analyze disk array reliability and availability. However, it is the transient errors during a disk access that could lead to catastrophic errors. We have developed detailed dependability models that take into account both the hard disk failures and transient errors. We have shown how various model parameters (such as byte error rate, probability of byte error on a disk access, etc.), which were not modeled in previous studies, significantly affect disk array dependability.

Our analysis confirms the results of earlier studies in that if data loss is the only failure mode considered (and catastrophic errors are ignored), then mirrored disk organization is the most reliable among various RAID organizations. However, if catastrophic errors are also considered, then RAID-3,4,5 organizations possess highest data integrity. This happens because various RAID organizations provide protection against data loss but do not provide protection from catastrophic failures. As far as catastrophic errors are concerned, the data integrity then depends simply upon the catastrophic failure rate and the way a disk array is organized for a given storage capacity (i.e., how many disks per group and how many groups). To improve data-integrity, more sophisticated ECC codes could be employed. To improve data-availability, redundancy can be increased. The reliability of RAID-2 is always worse than RAID-3,4,5 since it requires more disks but provides same protection (i.e., single disk failure in a group).

Whereas the MTTDL or data integrity remain virtually unchanged with increase in mean
recovery time, our analysis shows a large increase in degraded capacity time with increase in mean recovery time. This illustrates the need for using appropriate dependability measures to bring out the effects of certain model parameters which may remain hidden if other dependability measures are used.

Slight increase in byte error rate causes significant degradation in the reliability of disk arrays suggesting that this is one of the major bottlenecks in disk array dependability. Byte error rate could be reduced by periodic maintenance checking of the head and disk arms, so that chances of head misalignment are reduced. Disk array reliability does not scale as the dimensions of disk array are scaled to increase the storage capacity or to provide higher bandwidth. To maintain high reliability, more sophisticated ECC codes should be employed.

If reliability of support hardware (power supply, cooling hardware, array controller, host bus adaptor, etc.) is taken into account, then overall reliability of disk array decreases. Orthogonal placement of support hardware increases the overhead cost but significantly improves the reliability. Among different disk array architectures, mirrored disk architecture benefits the most from orthogonal placement support hardware. The overhead cost of orthogonal RAID-1 organization is the least among different RAID organizations while the gains in reliability are the most.

The utility of such analytic models will be enhanced if experimental data on various parameter values becomes available. These models can then be used for prediction of disk array reliability. We have sought to illustrate which parameters should be taken into account when modeling disk array dependability. The significance of these parameters is illustrated by comparison with earlier studies or with models which ignore these parameters.

Acknowledgements

We are grateful to Dr. Garth Gibson and Dr. Mark Holland of Carnegie Mellon University and Dr. Spencer Ng of IBM Almaden research center for several helpful comments. We would also like to thank Lorrie Tomek for useful suggestions.

References


Manish Malhotra received B. Tech. degree in Computer Science from Indian Institute of Technology, Delhi, India, in 1989, and M.S. degree in Computer Science from Duke University, Durham, NC, in 1990, where he is currently a Ph.D. candidate. He was a summer intern at AT&T Bell Labs, Holmdel, NJ, in 1990 and 1991. His research interests include dependability and performance modeling, numerical transient solution of Markov chains, and disk array systems.

Kishor S. Trivedi holds a Ph.D. in Computer Science from the University of Illinois, Urbana-Champaign. He is the author of the book entitled, Probability and Statistics with Reliability, Queuing and Computer Science Applications, published by Prentice-Hall. His research interests are in reliability and performance evaluation. He has lectured and published extensively on these topics. He is a Fellow of the Institute of Electrical and Electronics Engineers. He is a Professor of Electrical Engineering and Computer Science at Duke University, Durham, NC. He was an Editor of the IEEE Transactions on Computers from 1983–1987. He is a co-designer of HARP, SAVE, SHARPE and SPNP modeling packages.