Reliability Analysis of Disk Array Organizations by Considering Uncorrectable Bit Errors

Xuefeng Wu, Jie Li, and Hisao Kameda
Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba Science City, 305 Japan
{wuxf, lijie, kameda} @osdp.is.tsukuba.ac.jp

Abstract

In this paper, we present an analytic model to study the reliability of some important disk array organizations that have been proposed by others in the literature. These organizations are based on the combination of two options for the data layout, regular RAID-5 and block designs, and three alternatives for sparing, hot sparing, distributed sparing and parity sparing. Uncorrectable bit errors have big effects on reliability but are ignored in traditional reliability analysis of disk arrays. We consider both disk failures and uncorrectable bit errors in the model. The reliability of disk arrays is measured in terms of MTTDL (Mean Time To Data Loss). A unified formula of MTTDL has been derived for these disk array organizations. The MTTDLs of these disk array organizations are also compared using the analytic model.

1. Introduction

The increasing performance gap between CPUs and I/O systems threatens to create an I/O bottleneck that will limit overall system performance. Disk arrays or Redundant Arrays of Inexpensive Disks (RAID) [9] have been receiving increased attention for providing high performance and good data reliability to improve I/O systems. Disk arrays have been categorized into several levels [2]. *RAID-5* has a regular data layout that allows good performance in normal operations. In order to obtain a graceful performance degradation after a disk failure, data layouts that are based on *block designs* have been proposed [4],[8]. Spare space in disk arrays is provided for the failure recovery. How the spare space is organized for on-line failure recovery has been proposed in [9] (*hot sparing*), [6] (*distributed sparing*) and [8] (*parity sparing*). On-line spare disks are only employed in hot sparing policy but not in the other policies. Four disk array organizations that are based on the combination of two options for data layout, *regular RAID-5* and *block designs*, and the three alternatives for sparing, *hot sparing, parity sparing* and *distributed sparing*, have been proposed in [1]. The performance evaluation of these organizations has also been presented in the same paper. In this paper, we present an analytic model to study the reliability of these organizations.

The reliability of disk arrays is usually measured in terms of Mean Time To Data Loss (MTTDL). There are three relatively common ways [2] to lose data: 1. system crash followed by a disk failure; 2. double disk failures (We call this *DD data loss*); 3. disk failure followed by an uncorrectable bit error during the data reconstruction (we call this *DB data loss*). An UNcorrectable Bit Error (UNBE) is an incorrect sector read, as detected by an Error Correcting Code (ECC), which is uncorrectable by either ECC or retry [10]. After a disk failure, data on the surviving disks should be read to reconstruct the content of the failed disk. A DB Data loss occurs if an UNBE damages the necessary data.

Some studies have been conducted on the reliability analysis of disk arrays [5],[2],[3],[10]. UNBEs, however, have been ignored or have not been studied in detail in these analyses. More effort has been devoted to this problem in [11], which presented a analytic model to analyze the reliability of declustered-parity RAID (its data layouts are based on block designs).

In [11], RAID-5 is considered as a special case of declustered-parity RAID. It is assumed that on-line spare disks are employed for the failure recovery in the model. In this paper, we address the problem in a similar way but make important extension and improvement upon the previous work. Both disk failures and uncorrectable bit errors are considered in the model.

1060-9857/97 $10.00 © 1997 IEEE

The basic assumptions about them are the same as in [11]. However, the model has been amended to describe the disk array organizations unconsidered in [11]. A unified formula of MTTDL is derived for these disk array organizations. The analysis is simplified so that a single MTTDL value can be obtained instead of the optimistic and pessimistic estimates as in [11]. The limitation that the 'baseline' [7] recovery strategy is employed is also removed. Finally, the MTTDLs of these disk array organizations are compared using the analytic model.

This paper is organized as follows. The disk array organizations to be discussed are introduced in Sect. 2. The model is described in Sect. 3. Reliability analysis is developed in Sect. 4. The numerical study is given in Sect. 5. Conclusions come in Sect. 6.

2. Disk array organizations

A disk array consists of C primary disks holding *data stripe units* (or simply *data units*) and *parity stripe units* (or simply *parity units*) (a unit consists of an integral number of sectors). A parity unit is the XOR of associated bits on $G-1$ ($G \leq C$) data units. This parity unit and the $G-1$ data units form a *parity stripe*. These units are called *stripe-mate unit* (or simply *mate unit*) to one another. Each unit in a parity stripe is stored on a different disk of the array.

We consider the four important disk array organizations (Fig. 1) that have been proposed in [1]. There are T disks in a disk array system. M of them are primary disks. Every system employs the spare space of one disk for on-line failure recovery. In Fig. 1, 'd's, 'd1's and 'd2's are data units. 'p's, 'p1's and 'p2's are parity units. 's's are spare units (no data or parity). We call the four disk array organizations *the Hot-Sparing* (HS), *the Parity-Sparing* (without block designs) (PS), *the Block-Designs* (parity sparing with block designs) (BD) and *the Distributed-Sparing* (DS).

For hot-sparing organization, there is an on-line spare disk in the system besides the primary disks which form an array of regular RAID-5. $G = C = M = T-1$. After a disk failure, the lost data on the failed disk will be reconstructed onto the spare disk in the rebuild process. For the other three organizations, there is no on-line spare disk in the sytem, i.e., $M = T$.

For parity-sparing organization (parity sparing without block designs), the spare space is used to provide parity information and thus to reduce the parity stripe size. The scheme divides all the disks in the system into two small arrays of regular RAID-5. $G = C = M/2$ for every array (let T be an even integer). On a failure, the parities of the two arrays are merged to obtain a single array with almost doubled parity stripe size after the failure recovery (new par-

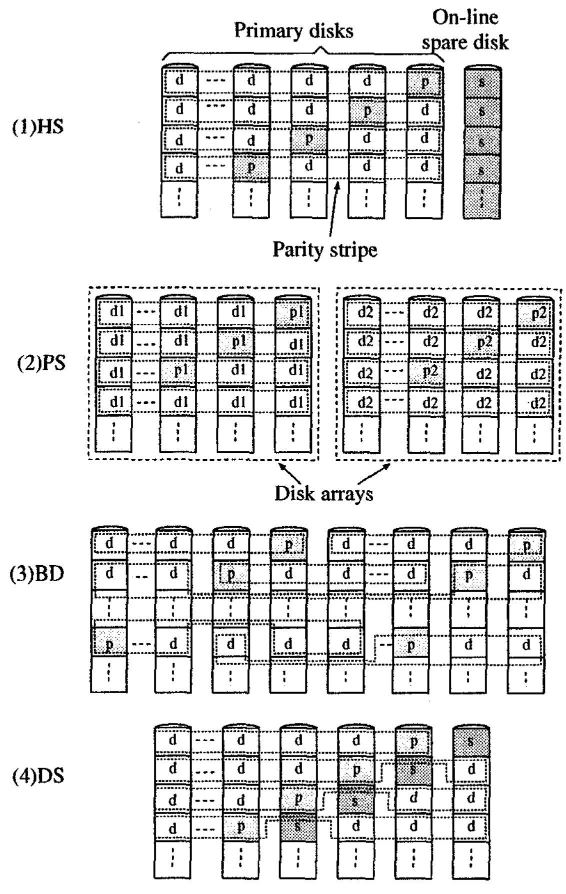

Figure 1. Disk array organizations.

ity stripe size becomes $T-1$). The data on the failed disk can been rebuilt onto the surplus space originally occupied by some of the old parities on the surviving disks. Only half of the disks in the system (those in the same small array, $G-1$ disks to be exact) have to be read for the reconstruction of all the units on the failed disk.

For block-designs (parity sparing with block design) organization, the parity stripes in the two arrays in above system have been mixed. The same as for parity-sparing organization, on a failure, the parity stripe size is to be doubled for failure recovery. And the surplus space originally occupied by some of the old parities are used for rebuilding the lost data on the failed disk. For the reconstruction of each unit on the failed disk, only half of the disks in the system have to be read each time (to read all $G-1$ mate units in the old parity stripe of this unit). For the reconstruction of all the units on the failed disk, however, differently from parity-sparing organization, all the surviving disks in the system have to be accessed. It is in this way that block-designs organization can provide better degraded performance than parity-sparing organization.

For distributed-sparing organization, the spare space on a disk is distributed on all the disks in the array instead of locating it on a single disk. On a failure, the lost data on the failed disk are rebuilt onto those spare units on the the surviving disks.

Note that after the failure recovery, the parity stripe size becomes $T-1$ for all the four organizations.

A system works in *normal duration* when all disks are functional. After a failure, the system works in *failure duration* till the rebuild of the failed disk is completed. The rebuild process can start as soon as the failure occurs, since on-line recovery scheme is employed. After the failure recovery, there is a *restoration duration* in which a new spare disk is brought into the system and then the system is restored to the same state as in normal duration.

In normal duration, a read request for a unit entails an actual read of the unit. A write request for a unit, however, entails four actual accesses: reading an old data from the target unit, reading an old parity from the mate parity unit for computing the new parity, writing the new data to the target unit, and writing the new parity to the mate parity unit.

When an UNBE is found during a actual read from some disk, all the mate units on other disks will be read to reconstruct the related unit onto a replacement unit on this disk. The time for rebuilding a unit for UNBEs is usually very short (e.g., less than a second) and the performance of system is hardly affected.

In failure duration, the system continuously satisfies requests for data while simultaneously reconstructing the content of the failed disk onto the replacement spare space.

In restoration duration, the data should be moved so that the original data layout before the failure can be restored for all the organizations except hot-sparing organization.

3. The reliability model

We take into account both DB data losses and DD data losses in the reliability analysis. Gibson has presented an excellent study of the DD data losses in [3]. Our work here is on the analytic study of DB data losses. A data loss may still occur in case a disk failure or another UNBE occurs during the rebuild of a bad sector for UNBE error (we call them *BD data loss* and *BB data loss* respectively). But the BD and BB data losses can be ignored in terms of MTTDL calculation, as shown in [12], an enhanced version of [11]. We assume that the workload of a disk array system in normal duration is evenly distributed over across all the primary disks in the system. The disk array system is only accessed by small reads/writes, i.e., one unit for an access request, which is a characteristic of on-line transaction processing. User read and write requests arrive independently with Poisson distributions. This process is suitable when the number of sources of request is relatively large. Assume that the requests are evenly distributed over all data.

Reading a disk is very unlikely to cause permanent errors. Most UNBEs are generated because data is incorrectly written or gradually damaged as the magnetic media ages [2]. We focus on the first case to simplify our model. It is assumed that UNBEs occur during writing and become evident during reads and that an UNBE-free unit will have UNBEs with probability A (a positive constant) after this unit has been accessed by a write. Factors such as imperfections on disk surface, weak heads, electronic noise, difficult data patterns, etc., may cause the UNBEs. UNBEs are mostly caused when two or more of these factors have a combined effect [5].

Under the above assumptions, UNBEs that already existed in a unit will always be detected before new data can be actually written to this unit.

A DB data loss occurs due to an UNBE during the reconstruction of the content of the failed disk – this UNBE has destroyed the data, say some unit, on some surviving disk, which is necessary for this reconstruction. Under above assumptions, there are two possible cases for the occurrence of this UNBE. This UNBE might have occurred before the failure on one of the surviving disks due to a write to the related unit without having been discovered. Or this UNBE might have occurred due to a write after the failure (in failure duration). We deal with the two cases differently in the reliability analysis.

For DD data losses, we only consider independent disk failures. The main data losses occur when another a second disk fails before the rebuild is completed; therefore, if the length of restoration duration is relatively short, then the data losses due to disk failures in this period will not pose a problem to the reliability since the system can still tolerate one disk failure. Long restoration duration can be avoided by employing cold standby disks [3] in actual systems. We do not consider issues associated with the data losses in restoration duration.

In the system of hot-sparing organization, we assume that an on-line spare disk is always available when a disk failure occurs, so the failure of the spare disk is not considered in our reliability analysis. In an actual system, a special background failure-detecting scheme for the on-line spare disk should be employed. Otherwise the failure of on-line spare disk may go unnoticed which will threaten the system with data loss [1].

For parity-sparing organization, we consider the DD data losses that are due to any two failures among the $M/2$ disks within one of the two small arrays. For the other organizations, the DD data losses are due to any two failures among the primary disks.

4. Reliability analysis

First we give the notations used in the analysis as follows:

PR_{db}: probability that a data loss occurs due to an UNBE during the data reconstruction after a disk failure.

$MTTF_{disk}$: Mean Time To Failure for a single disk.

T_{rb}: rebuild time of the failed disk.

A: probability that an UNBE-free unit will have UNBEs after the unit has been accessed by a write.

(The following are the parameters for the system considered in normal duration)

X: rate of user data requests to the system. (units/sec)

M: number of primary disks in the system.

C: number of disks in a disk array.

G: number of units in a parity stripe.

λ_w: arrival rate of user write requests to a disk. (units/sec)

λ_r: arrival rate of user read requests to a disk. (units/sec)

F_w: fraction of user data requests that are writes.

N_d: number of data units in a disk.

N_p: number of parity units in a disk.

The formulas for calculating the MTTDLs of the disk array systems are unified by carefully defining and using the parameters in the formulas, so that the formulas for all the four organizations have the same form. Thus, the following analysis is also applicable to any one of the four disk array organizations.

Note that we have the following relations:

$\lambda_w = F_w * (\frac{X}{M}), \quad \lambda_r = (1 - F_w) * (\frac{X}{M}),$
$F_w = \frac{\lambda_w}{\lambda_w + \lambda_r}, \quad N_d : N_p = (G-1) : 1.$

Formulas (1), (2), and (3) for calculating the MTTDLs of a disk array system are shown as follows. They are simple extensions of the formulas for RAID-5 presented in [2]. These formulas are obtained by applying the results presented in [9] and [3].

$$MT = \frac{MT_{dd} * MT_{db}}{MT_{dd} + MT_{db}} \tag{1}$$

where

$$MT_{dd} = \frac{MTTF_{disk}^2}{(M/C)C(C-1)T_{rb}} \tag{2}$$

$$MT_{db} = \frac{MTTF_{disk}}{M * PR_{db}} \tag{3}$$

Here, MT is the MTTDL of the system that is calculated by taking account of both DD data losses and DB data losses. MT_{dd} is the MTTDL that is calculated by only taking into account DD data losses (independent disk failures). MT_{db} is the MTTDL that is calculated by only taking account of DB data losses. It is assumed that the disk failure process is exponentially distributed with mean $MTTF_{disk}$. M/C is the number of arrays in the system. M/C is 2 for parity-sparing organization and 1 for the other organizations.

In the formulas, C, M, $MTTF_{disk}$ and T_{rb} are supposed to be known. The following work is to calculate the PR_{db} in (3).

Let $PR_{ok} = 1 - PR_{db}$. Thus, PR_{ok} is the probability that the rebuild of the failed disk can be successfully completed without the trouble of UNBEs. Therefore, we have the expression of PR_{ok} as follows, which means the probability that every unit on the failed disk can be successfully rebuilt without the trouble of UNBEs when reading their mate units on the surviving disks.

$$PR_{ok} = \Pi_{i \in (S_d \cup S_p)} PU(i) \tag{4}$$
$$(PR_{db} \equiv 1 - \Pi_{i \in (S_d \cup S_p)} PU(i))$$

i: index of a unit on the failed disk.

S_d: set of the data units on the failed disk.

S_p: set of the parity units on the failed disk.

$PU(i)$: probability that for unit i, no UNBE has been discovered by user data requests in failure duration in all its mate units before rebuilding unit i (so that the system can survive till the rebuild of unit i starts) and no UNBE is discovered in all the *necessary* mate units when these units are read for rebuilding unit i (so that the data reconstruction for rebuilding unit i can be completed successfully).

Rewrite PR_{ok} in (4),

$$PR_{ok} = PU(S_d) * PU(S_p) \tag{5}$$

where

$$PU(S_d) = \Pi_{i \in S_d} PU(i), \; PU(S_p) = \Pi_{i \in S_p} PU(i).$$

To simplify the analysis, we assume that for rebuilding each unit on the failed disk *all* its mate units on the surviving disks are necessary. For rebuilding a data unit on the failed disk, its mate parity unit may be unnecessary sometimes when 'baseline' recovery strategy [7] is employed, as stated in [11]. It can been seen from the numerical results presented in [11] that ignorance of this particular case is reasonable since the related optimistic estimate and the pessimistic estimate of the MTTDL are close to each other.

With this assumption, we can redefine $PU(i)$ in (5) as:

$$PU(i) = PU_0(i) * PU_f(i) \qquad (i \in S_d \cup S_p)$$

$PU_0(i)$: probability that no UNBEs exist in all the

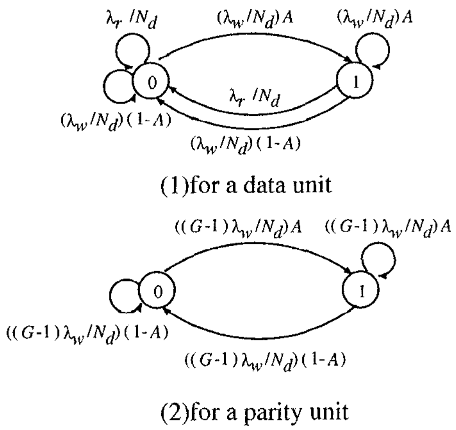

Figure 2. Model of occurrences of UNBEs in normal duration.

mate units of unit i at the moment when a disk failure occurs.
$PU_f(i)$: probability that before all the mate units of unit i are read for the rebuild, all the writes to these units after the disk failure (in failure duration) have not caused any UNBEs.

We call the UNBEs that exist on the surviving disks at the moment when the failure occurs *old UNBEs*, and call the UNBEs that are caused by the writes after the failure (in failure duration) *new UNBEs*. So the condition for a unit on the failed disk to be successfully rebuilt is that before the rebuild of this unit is completed, all its mate units on the surviving disks should be free of old UNBEs and new UNBEs.

Rewrite $PU(S_d)$ for data units in (5):

$$PU(S_d) = PU_0(S_d) * PU_f(S_d) \tag{6}$$

where

$$PU_0(S_d) = \Pi_{i \in S_d} PU_0(i), PU_f(S_d) = \Pi_{i \in S_d} PU_f(i).$$

Rewrite $PU(S_p)$ for parity units in (5):

$$PU(S_p) = PU_0(S_p) * PU_f(S_p) \tag{7}$$

where

$$PU_0(S_p) = \Pi_{i \in S_p} PU_0(i), PU_f(S_p) = \Pi_{i \in S_p} PU_f(i).$$

4.1. Old UNBEs

To derive the expression of PU_0, only "old UNBEs" have to be considered. This is to analyze the occurrences of UNBEs in the disk array in normal duration. We develop the models of Markov chain to obtain the probability that a unit is free of UNBEs in normal duration shown in Fig. 2.

There are two states in the models of Markov chain. State 0 represents the state that a unit does not have UNBEs in normal duration. State 1 represents the state that a unit has UNBEs in normal duration. Reads and writes requests to this unit lead to possible state transitions. The rate of read requests to a data unit is λ_r/N_d. The rate of write requests to a data unit is λ_w/N_d. The rate of implicit write requests to a parity unit is $(G-1)\lambda_w/N_d$ since a write request arriving at one of its $(G-1)$ mate data units will entail a write request arriving at this parity unit. Note that a read request arriving at a unit (data or parity unit) entails an actual read of this unit. A write request arriving at a unit, however, results in two actual accesses to this unit - reading old value from this unit (old-value-reading), and then writing new value to this unit (new-value-writing).

Consider the model for data units (Fig. 2(1)). When a data unit does not have UNBEs (state 0), read requests do not cause UNBEs to it (trivial to-itself transition at rate λ_r/N_d from state 0). A write request may cause UNBEs on this unit (by the new-data-writing) (transition to state 1 with the rate of $(\lambda_w/N_d)A$ and trivial to-itself transition at rate $(\lambda_w/N_d)(1-A)$). When the unit has UNBEs (state 1), a read request will lead to the discovery of possibly existing UNBEs, and then the rebuild of this unit onto a replacement unit (transition to state 0 at rate λ_r/N_d). A write request will lead to the discovery of possibly existing UNBEs by the old-data-reading and UNBEs may still occur during the new-value-writing to the replacement unit (transition to state 0 at rate $(\lambda_w/N_d)(1-A)$ and to-itself transition at rate $(\lambda_w/N_d)A$).

It is assumed that it is always possible to find a replacement unit for the unit with UNBEs. The entailed reduction of available data volume on the disk is ignored. The rebuild time for the bad unit is also ignored because it is very short compared with the average interval between the access requests to this unit.

Following steady state equations are derived from the model:
$PS0$: probability that a data unit does not have UNBEs in normal duration.
$PS1$: probability that the data unit has UNBEs in normal duration.

$$PS0 + PS1 = 1$$
$$\left(\frac{\lambda_w}{N_d}\right) A * PS0 = \left(\frac{\lambda_r}{N_d} + \frac{\lambda_w}{N_d}(1-A)\right) * PS1$$

Then we have:

$$PS0 = 1 - F_w * A, \quad PS1 = F_w * A.$$

The model for parity units is developed and used in the same way:
$PS0'$: probability that a parity unit does not have UNBEs in normal duration.
$PS1'$: probability that the parity unit has UNBEs in normal duration.

$PS0' + PS1' = 1$

$$\left(\frac{(G-1)\lambda_w}{N_d}\right) A * PS0' = \left(\frac{(G-1)\lambda_w}{N_d}\right)(1-A) * PS1'$$

Then we have:

$$PS0' = 1 - A, \quad PS1' = A.$$

Therefore, according to the definitions of $PU_0(i)$ and $PU_0(S_d)$, we have:

$$PU_0(i) = PS0^{G-2} * PS0' \qquad (i \in S_d)$$

$$PU_0(S_d) = (PS0^{G-2} * PS0')^{N_d} \tag{8}$$

Similarly, for $PU_0(i)$ $(i \in S_p)$ and $PU_0(S_p)$, we have:

$$PU_0(i) = PS0^{G-1} \qquad (i \in S_p)$$

$$PU_0(S_p) = (PS0^{G-1})^{N_p} \tag{9}$$

4.2. New UNBEs

To derive the expression of PU_f, only "new UNBEs" have to be considered. This is to analyze the occurrences of UNBEs after the disk failure (in failure duration). Assume that a unit i on the failed disk is rebuilt at moment $t = t_i$ and that the failure occurs at moment $t = 0$. The rate of writes to a data unit during the time interval t_i is the same as in any other (normal) duration (the rate is λ_w/N_d). Before a data unit on the failed disk is rebuilt, the rate of implicit writes to its mate parity unit on some surviving disk is $(G-2)\lambda_w/N_d$ since a write to one of the $(G-2)$ mate data units (on the surviving disks) entails a write to this parity unit.

Data units

For a data unit i on the failed disk, according to the definition of $PU_f(i)$, we have:

$$PU_f(i) = (1-A)^{MD(i)}$$

where

$$MD(i) = (G-2) * \frac{\lambda_w}{N_d} * t_i + \frac{(G-2)\lambda_w}{N_d} * t_i$$
$$= \left(2(G-2)\left(\frac{\lambda_w}{N_d}\right)\right) * t_i.$$

$MD(i)$ is the number of writes to all the mate units of unit i in failure duration before these units are read for rebuilding unit i. Unfortunately, it is difficult if not impossible to obtain the t_i $(i \in S_d)$ analytically. We conservatively let all t_is be T_{rb} and obtain the estimate of $PU_f(S_d)$ according to its definition in (6):

$$PU_f(S_d) = (1-A)^{MDS} \tag{10}$$

where

$$MDS = \sum_{i \in S_d} MD(i)$$
$$= 2(G-2)\lambda_w T_{rb}.$$

So far, formula of $PU(S_d)$ has been completely derived (according to (6), (8) and (10)).

Parity units

Similarly, we also derive the expression for $PU_f(S_p)$ as:

$$PU_f(S_p) = (1-A)^{MPS} \tag{11}$$

where

$$MPS = \sum_{i \in S_p} MP(i)$$
$$= (G-1)\lambda_w T_{rb}\left(\frac{N_p}{N_d}\right).$$

$MP(i)$ is the number of writes to all the mate units of unit i in failure duration before these units are read for rebuilding unit i. Thus formula of $PU(S_p)$ has been completely derived (according to (7), (9) and (11)). So far, the formula derivation of PR_{db} has been completed.

In deriving $PU_f(S_d)$ and $PU_f(S_p)$, we can also let all t_is be $T_{rb}/2$ to make the approximation. The numerical study we did shows that the comparison study of the four disk array organizations in this paper is not affected though the calculated MTTDLs for these organizations all become somewhat larger in this case.

5. Numerical results

In this numerical study, major parameters are taken from [1] which studied the performance of the four disk array organizations by a simulation model. The parameters listed below are used unless otherwise specified:

Normal request rates (X): 50, 100, 200 units/second.
Fraction of write (F_w): 30%.
Geometry: 1258 cylinders/disk, 14 tracks/cylinder, 52 sectors/track.
Unit size: 1 track
Sector size: 512 bytes
Data volume of a disk (S): 17612 units (1258*14).
System size (T): 16 disks.
Recovery strategy: 'baseline'.
Mean Time To Failure for a disk ($MTTF_{disk}$): 200,000 hours.
Error Probability (A): $52/(2.4 * 10^{10})$.

Table 1. Parameters for the four disk array organizations

	M	C	G	N_d	N_p
HS	15	15	15	$S*(G-1)/G$	S/G
PS	16	8	8	$S*(G-1)/G$	S/G
BD	16	16	8	$S*(G-1)/G$	S/G
DS	16	16	15	$S*(G-1)/M$	S/M

$MTTF_{disk}$ is given a typical value of 200,000 hours [2]. Most disks cite uncorrectable bit error rates of 1

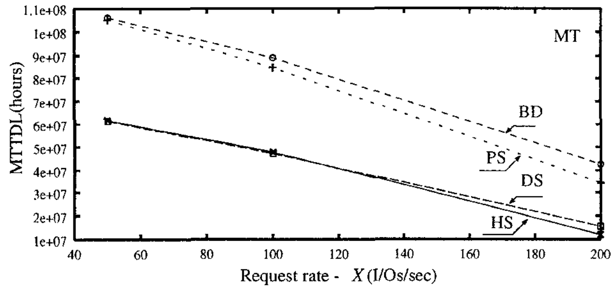

Figure 3. MTTDLs of the disk array systems.

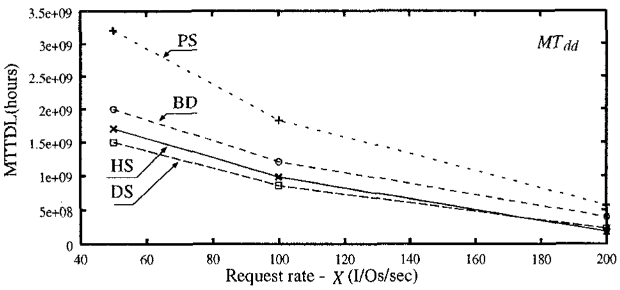

Figure 4. MTTDLs when only DD data losses are considered.

error per 10^{14} bits read, which are equivalent to 1 error per $2.4*10^{10}$ sector read [2]. So we give error probability A a default value of $52\ /(2.4*10^{10})$ (a unit consists of 52 sectors). The rebuild times are read directly from the Fig. 6 in [1].

MTTDLs as functions of request rate X are plotted in Fig. 3. Both DD data losses and DB data losses are taken into account in the calculation. It is shown that the reliability of a disk array system becomes worse as the request rate increases, since the rebuild time becomes longer.

In Fig. 4, only DD data losses are taken into account for contrast. It can been seen that the MTTDLs of the disk array systems shown in Fig. 3 are much shorter than that shown in Fig. 4. The UNBEs have big effects on the reliability of the systems.

In Fig. 3, the system of hot-sparing organization and the system of distributed-sparing organization have almost the same MTTDLs, which are much shorter than that of the system of parity-sparing organization and the system of block-design organization. The systems of hot-sparing organization and distributed-sparing organization have bigger parity stripe sizes which make them much more vulnerable and accounts for DB data losses.

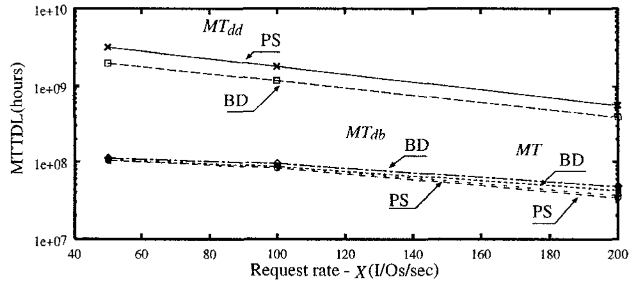

Figure 5. Comparison of the block-designs organization and the parity-sparing organization.

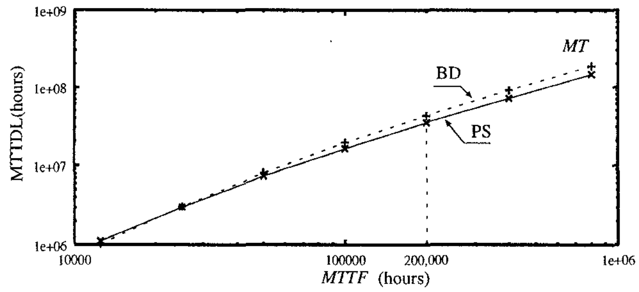

Figure 6. Effect of $MTTF_{disk}$ on the MTTDLs.

Block-designs organization provides the highest reliability among the four organizations. However, it is much less reliable than the parity-sparing when only DD data losses are considered as shown in Fig. 4. The reason behind this is revealed in Fig. 5. In Fig. 5, MT_{dd}s, MT_{db}s and MTs as the functions of request rate X only for the system of block-designs organization and parity-sparing organization are depicted. The system of block-designs organization has longer MT_{db} and shorter MT_{dd}. However, as the MT_{db} of a system is much shorter than the MT_{dd}, their combination MT has mainly be determined by the MT_{db}. In this case, DB data losses dominates the data losses, though only DD data losses are traditionally considered in the reliability analysis of disk arrays.

The remainder of this section is concerned only with block-designs and parity-sparing organizations because they are much more reliable than hot-sparing and distributed-sparing organizations. The MTs of block-designs and parity-sparing organization increase with the decrease of $MTTF_{disk}$ (Fig. 6) or the increase of error probability A (Fig. 7). Request rates are fixed to 200 units/second in the calculation in figures 6 and 7. It is shown that block-designs organiza-

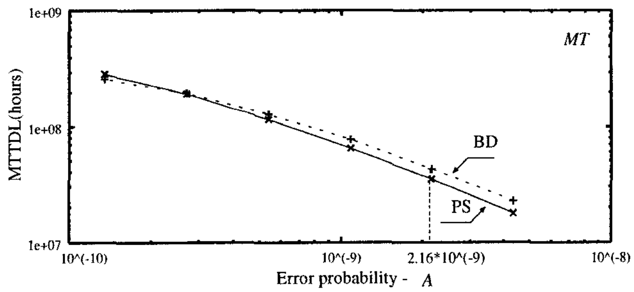

Figure 7. Effect of error probability A on the MTTDLs.

tion has a shorter MT than parity-sparing organization only when the $MTTF_{disk}$ or the error probability A becomes much smaller than the respective default value.

In the above calculation, the 'baseline' rebuild strategy is employed for the systems. We also made the calculation for another rebuild strategy, the 'minimal operation' strategy proposed in [1]. Similar results are obtained.

The systems of 16 disks ($T = 16$) are considered as medium size disk array systems. For the systems of 7 disks ($T = 7$), the small size disk array systems considered in [1], we also calculated their reliability. Similar approach is employed in this case and the formulas become more complicated since T is not an even number. Similar results are also obtained.

6. Conclusions

In this paper, we presented an analytic model to study the reliability of some important disk array organizations. Both disk failures and uncorrectable bit errors are considered in the model. A unified formula of MTTDL for these disk array organizations has been derived.

We showed that uncorrectable bit errors have big effects on the reliability of the disk arrays. DB data losses might dominate the data losses of the disk array systems though only DD data losses are traditionally considered. For the four important organizations considered in this paper, it is shown that block-designs organization and parity-sparing organization are much preferable to hot-sparing organization and distributed-sparing organization on reliability for smaller parity stripe size. Block-designs organization provides higher reliability than parity-sparing organization when the request rate of disk access is high. For the system of block-designs organization in this case, the merit of shorter rebuild time can make up the dismerit of larger array size.

References

[1] J.Chandy and A.L.Narasimha Reddy, "Failure evaluation of disk array organizations", *Proc. 13th Int'l Conf. Distributed Computing Systems*, 319-326, 1993.

[2] P.M.Chen, E.K.Lee, G.A.Gibson, R.Katz, and D.A.Patterson, "RAID:high-performance, reliable secondary storage", *ACM Computing Surveys*, Vol.26, No.2, 145-185, June 1994.

[3] G.A.Gibson, Redundant Disk Arrays: Reliable, Parallel Secondary Storage, *ACM Distinguished Dissertation Series*, MIT Press, Cambridge,1992.

[4] M.Holland and G.A.Gibson, "Parity declustering for continuous operation in redundant disk arrays", *Proc. ASPLOS V-10/92/MA,USA*, 23-35, 1992.

[5] M.Malhotra and K.S.Trivedi, "Data integrity analysis of disk array systems with analytic modeling of coverage", *Performance Evaluation*, Vol.22, 111-133, 1995.

[6] J.Menon and R.Mattson, "Comparison of sparing alternatives for disk arrays", *Proc. Int'l Symp. Comp. Architecture*, 1992.

[7] R.R.Muntz and J.C.S.Lui, "Performance analysis of disk arrays under failure", *Proc.16th VLDB Conference*, 162-173, 1990.

[8] A.L.Narasimha Reddy and P.Banerjee, "Gracefully degradable disk arrays", *Proc. FTCS-21*, 1991.

[9] D.A.Patterson, G.A.Gibson, and R.H.Katz, "A case for Redundant Arrays of Inexpensive Disks (RAID)", *Proc. ACM SIGMOD-88*, pp.109-116, 1988.

[10] M.Schulze, G.A.Gibson, R.Katz, and D.A.Patterson, "How reliable is a RAID", *Proc. IEEE COMPCON-89*, 118-123, 1989.

[11] X.Wu, J.Li and H. Kameda, "Reliability analysis of declustered-parity RAID focusing on uncorrectable bit errors", *Proc. Int'l Conf. Computer Architecture*, (belonging to *1996 Int'l Computer Symposium*), Taiwan, 1-6, 1996.

[12] X.Wu, J.Li and H. Kameda, "Reliability modeling of declustered-parity RAID considering uncorrectable bit errors", *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E80-A, No.8, August 1997. (to appear)