Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Reliability Analysis of Declustered-parity RAID Focusing on Uncorrectable Bit Errors

Xuefeng Wu, Jie Li and Hisao Kameda
Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba Science City, Ibaraki 305, Japan

Abstract

UNcorrectable Bit Errors (UNBEs) have been ignored or have not been studied in detail in existing analysis of the reliability of RAID. We present an analytic model to study the effect of UNBEs on the reliability of declustered-parity RAID. The Mean Time To Data Loss (MTTDL) of the RAID is calculated by taking into account both DB data loss and the data loss caused by double independent disk failures (we call this DD data loss). On the basis of our numerical analysis, we discuss how such MTTDL depends on the number of units in a parity stripe and the rebuild time of a failed disk.

1 Introduction

Redundant Array of Inexpensive Disks (RAID) has been proposed to bridge the gap between high CPU performance and low I/O performance with maintaining high data reliability at the same time [7]. RAID-5 is among the most popular ones. A variation of RAID-5 called *declustered-parity RAID* [2] (it is also called *clustered RAID* in [5]) provide better performance in the presence of failed disks.

Reliability is an important issue in the evaluation of RAID systems. We study the reliability of declustered-parity RAID in terms of Mean Time To Data Loss(MTTDL). There are three relatively common ways [2] to lose data in declustered-parity RAID: 1.system crash followed by a disk failure; 2.double disk failure (We call this *DD data loss*); 3.disk failure followed by an uncorrectable bit error during reconstruction (we call this *DB data loss*). An UNcorrectable Bit Error (UNBE) is a sector read incorrectly, as detected by an Error Correcting Code (ECC), which is uncorrectable by either ECC or retry [6]. After a disk in the array failed, data on the surviving disks should be read to reconstruct the content of the failed disk. DB Data loss occurs if an UNBE damages the necessary data.

Many studies have been conducted in analysing the reliability of RAID [1][2][3][6]. UNBEs, however, have been ignored or have not been studied in detail in existing analysis of the reliability of RAID. In this paper, we especially analyse the reliability of declustered-parity RAID focusing on UNBEs.

A new analytic model is provided to study the reliability of declustered-parity RAID in detail. By using the model, the optimistic and pessimistic estimates of the probability that data loss occurs due to an UNBE during the reconstruction after a disk failed (so called DB data loss) are obtained. MTTDL is calculated by taking into account both DD data loss and DB data loss. The optimistic and pessimistic estimates of the MTTDL are then obtained according to the optimistic and pessimistic estimates of DB data loss probability.

On the basis of our numerical analysis, we study how MTTDL depend on the number of units in a parity stripe and rebuild time. It is shown that the MTTDL depends largely on the number of units in a parity stripe. The MTTDL becomes smaller when parity stripe size is increased. And such MTTDL is less sensitive to the rebuild time of the failed disk than the MTTDL that only takes into account DD data loss in the traditional calculation.

The model is described in section 2. Reliability analysis is developed in section 3. The numerical study is given in section 4. Conclusions come in section 5.

2 Model of declustered-parity RAID

A array of declustered-parity RAID 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 $G-1$ 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*) each other. Each unit in a parity stripe is stored on a different disk of the array ($G \leq C$). RAID-5 can be taken as the special case that $G = C$.

When there is no failed disk in a disk array, we say that the disk array works in *normal duration*. After a disk fails, the disk array works in *failure duration* till the rebuild of the failed disk is completed.

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 old data from the target unit, reading old parity from the mate parity unit for computing the new parity, and writing the new data to the target unit and the new parity to the mate parity unit.

We assume that the workload is evenly distributed over across all the disks in the array. The disk array is accessed by small reads/writes, i.e., one unit for an access request, which is a characteristics 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 high. Assume that the requests are evenly distributed over all data.

When an UNBE is found during a 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.

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 writes 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 [1].

For on-line failure recovery scheme is employed, we assume that once a disk in the array has failed, the rebuild process of the failed disk can start soon by using an on-line spare disk. In failure duration, The disk array continuously satisfies requests for data while simultaneously reconstructing the content of the failed disk and copying them onto the replacement disk. Assume that once the destroyed data has been successfully reconstructed it can always be copied correctly onto the replacement disk.

Data loss occurs mainly when another disk fails in failure duration (so called DD data loss) or an UNBE is discovered during the data reconstruction (so called DB data loss). [3] presents an excellent study of DD data loss. Our study focuses on DB data loss.

There are many schemes [4][5] that may be employed to deal with the requests for the data on the failed disk and to proceed system rebuild process. We consider the so-called *user-writes* algorithm (which correspondents to the *Baseline copy* procedure in [5]) with *single-thread* [4] rebuild process in our analysis. System rebuild process simply sequentially reads units from the failed disk. Such read requests results in reading the mate units on the surviving disks to reconstruct the target units and then writing them to the spare disk. A user write request for a unit on the failed disk that has not been rebuilt on the spare disk will also result in the rebuild of this unit - writing the new data to the spare disk, reading the mate units on the surviving disks to reconstruct the new parity and then writing the new parity to the mate parity unit on the surviving disk. In this case, all the mate units except the mate parity unit should be free of UNBEs to successfully reconstruct the new parity. We call the parity unit *uncared-parity-unit* for the rebuild of its mate data unit on the failed disk (assume that this parity unit still is checked by read operation before new value is written to this unit).

3 Reliability analysis

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

PR_{db}: probability that data loss occurs due to an UNBE during the reconstruction after a disk in the array has failed.
$MTTF_{disk}$: 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.
X: rate of user requests to the disk array (units/sec).
C: number of disks in the disk array.
G: number of units in a parity stripe.
λ_w: rate of write requests to a disk in the array (units/sec).
λ_r: rate of read requests to a disk in the array (units/sec).
F_w: fraction of user requests that are writes.

N_d : number of data units in a disk.
N_p : number of parity units in a disk.

Note that we have the following relations: $\lambda_w = F_w * (\frac{X}{C})$, $\lambda_r = (1 - F_w) * (\frac{X}{C})$, $F_w = \frac{\lambda_w}{\lambda_w + \lambda_r}$, $N_d : N_p = (G-1) : 1$.

Formulas for calculating the MTTDL of declustered-parity RAID are as follows. They are a simple extension of the formulas for RAID-5 in [2]:

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

where

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

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

Here, MT is the MTTDL of the RAID that is calculated by taking into account both DD data loss and DB data loss. MT_{dd} is the MTTDL that is calculated by only taking into account DD data loss (independent disk failures). MT_{db} is the MTTDL that is calculated by only taking account of DB data loss. Both $MTTF_{disk}$ and T_{rb} are assumed to be exponential.

In the formulas, C, $MTTF_{disk}$ and T_{rb} are supposed to be known. We focus on the effect of UNBEs on the reliability of declustered-parity RAID. This is to calculate the PR_{db} in (3).

Let $PR_{ok} = 1 - PR_{db}$. PR_{ok} is the probability that no UNBE has been found in rebuilding the failed disk. 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.

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

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

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 no UNBE has been discovered by normal user requests in all the mate units of unit i just before they are read for the rebuild of unit i (so that the disk array can survive till the rebuild of unit i starts) and there is no UNBE on the related mate units when these units are read for the rebuild (so that the reconstruction for the rebuild of unit i can be successfully completed).

Rewrite PR_{ok} in (4),

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

where

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

To simplify the analysis, we first assume that all its mate units on the surviving disks are necessary for the rebuild of each unit on the failed disk, which does not take into account the uncared-parity-units for the rebuild of some data units on the failed disk. We will reconsider the uncared-parity-units problem in section 3.3. With this assumption, we can redefine $PU(i)$ 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 mate units of unit i at the moment when the disk fails.
$PU_f(i)$: probability that until all the mate units of unit i are read for the rebuild, all the writes to these units in failure duration have not caused any UNBEs.

We call the UNBEs that exist on the surviving disks at the moment when the disk fails *old UNBEs*, and call the UNBEs that are caused by the writes 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 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

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

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

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

where

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

3.1 Old UNBEs

To derive the expression of PU_0, only 'old UNBEs' have to be considered. This is to analyse the occurrence of UNBEs in the disk array in normal duration. We have developed the models of Markov chain to obtain the probabilities that a unit is free of UNBE in normal duration. Define $PS0$ and $PS0'$ as:
$PS0$: probability that a data unit does not have

UNBEs in normal duration.

$PS0'$:probability that a parity unit does not have UNBEs in normal duration.

They have been obtained according to the models as:

$$PS0 = 1 - F_w * A, \qquad PS0' = 1 - A$$

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:

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

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

3.2 New UNBEs

To derive the expression of PU_f, only 'new UNBEs' have to be considered. This is to analyse the occurrence of UNBEs in the disk array in failure duration. Assume that a unit i on the failed disk is rebuilt at moment $t = t_i$ and that the disk fails at moment t=0, so its mate units will be read for build at almost the same moment according to the recovery scheme we have taken into account. The rate of writes to a data unit in failure duration is the same as in 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 a surviving disk is $(G-2)\lambda_w/N_d$ since any a write to one of the $(G-2)$ mate data units (on the surviving disks) of unit i 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

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

$$= (2(G-2)(\frac{\lambda_w}{N_d})) * t_i$$

$MD(i)$ is the number of writes to the mate units of unit i in failure duration before these units are read for the rebuild. Unfortunately, it is difficult if not impossible to obtain the t_i $(i \in S_d)$ analytically. We let the average of t_is $(i \in S_d)$ to be $T_{rb}/2$ 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

$$MDS = \sum_{i \in S_d} MD(i)$$

$$= (2(G-2)(\frac{\lambda_w}{N_d}))(T_{rb}/2)N_d.$$

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

Parity units

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

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

where

$$MP(i) = (G-1) * (\frac{\lambda_w}{N_d}) * t_i$$

$$= ((G-1)(\frac{\lambda_w}{N_d})) * t_i.$$

$MP(i)$ is the number of writes to the mate units of unit i in failure duration before these units are read for the rebuild. Similarly, we let the average of t_is $(i \in S_p)$ to be $T_{rb}/2$ and obtain the estimate of $PU_f(S_p)$ (according to (7)):

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

where

$$MPS = \sum_{i \in S_p} MP(i)$$

$$= ((G-1)(\frac{\lambda_w}{N_d}))(T_{rb}/2)N_p.$$

Thus formula of $PU(S_p)$ has been completely derived (according to (7), (9) and (11)).

3.3 Pessimistic and optimistic estimates

The Estimate of $PU(S_d)$ developed in last section is actually a Pessimistic Estimate (i.e. $EP(PU(S_d))$) for 'user-writes' recovery scheme, since UNBEs on so called uncared-parity-units (when user write requests to the failed disk lead to the rebuild of the data units on the failed disk) will not affect the reconstruction for the related rebuild. So, we can have the $EP(PU(S_d))$ as follows :

$$EP(PU(S_d)) = EP(PU_0(S_d)) * EP(PU_f(S_d)) \tag{12}$$

where (according to (6), (8) and (10)),

$$EP(PU_0(S_d)) = (PS0^{G-2}PS0')^{N_d}$$

$$EP(PU_f(S_d)) = (1-A)^{(2(G-2)(\frac{\lambda_w}{N_d}))(T_{rb}/2)N_d}.$$

The the maximum number of of uncared-parity-units is obtained by assuming that every user write request arriving at the failed disk in failure duration contributes to the rebuild of the failed disk: $Z = \lambda_w * T_{rb}$. Assuming that both old UNBEs and new UNBEs on as much as Z uncared-parity-units do not cause data loss, we derive the Optimistic Estimate of $PU(S_d)$ (i.e. $EO(PU(S_d))$:

$$EO(PU(S_d)) = EO(PU_0(S_d)) * EO(PU_f(S_d)) \quad (13)$$

where

$$EO(PU_0(S_d)) = (PS0^{G-2})^{N_d} * PS0'^{N_d - Z}$$

$$EO(PU_f(S_d)) = (1-A)^{(2(G-2)(\frac{\lambda_w}{N_d}))(T_{rb}/2)(N_d - Z/2)}.$$

With above results we make the optimistic and pessimistic estimates of PR_{db} as (according to (4) and (5)):

$$EP(PR_{db}) = 1 - EP(PU(S_d)) * PU(S_p) \quad (14)$$

$$EO(PR_{db}) = 1 - EO(PU(S_d)) * PU(S_p) \quad (15)$$

Note that $EP(PR_{db}) \geq EO(PR_{db})$ whereas $EP(PU(S_d)) \leq EO(PU(S_d))$ (according to (12), (13), (4) and (5)). The pessimistic estimate of MTTDL (i.e. $EP(MT)$) with respect to the pessimistic estimate of PR_{db} is obtained by substituting the PR_{db} in (3) with the $EP(PR_{db})$ in (14). The optimistic estimate of MTTDL (i.e. $EO(MT)$) with respect to the optimistic estimate of PR_{db} is obtained by substituting the PR_{db} in (3) with the $EO(PR_{db})$ in (15). Then we have $EP(MT_{db}) \leq EO(MT_{db})$ and $EP(MT) \leq EO(MT)$ (according to (3) and (1)).

4 Numerical results

In this numerical study, major parameters are taken from [4] which studies the performance of declustered-parity RAID with a simulation model. The parameters listed as follows are used unless otherwise specified:

User access rate(X): 105 units/second.
Fraction of write(F_w): 50%.
Geometry: 949 cylinders, 14 heads, 48 sectors/tracks.
Unit size: 8 sectors
Sector size: 512 bytes

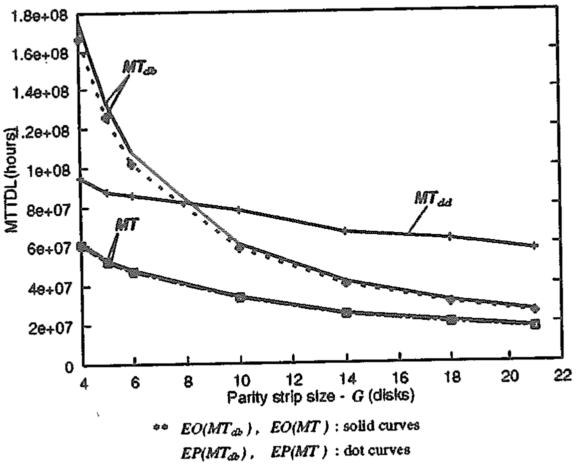

Figure 1: Effect of parity stripe size (C=21)

Data volume of a disk(S): 79716 units (949*14*48/8).
Array size (C): 21 disks.
parity stripe size (G): 4,5,6,10,14,18,21.
Recovery scheme: user-writes.
Mean Time To Failure for a disk ($MTTF_{disk}$): 200,000 hours.
Error Probability (A): $1/(3 * 10^9)$.

Let $N_d = ((G-1)/G) * S$, $N_p = (1/G) * S$. $MTTF_{disk}$ is given a typical value of 200,000 hours [2]. Most disks provides the bit error rate of 1 error per 10^{14} bits read, which is equivalent to 1 error per $2.4 * 10^{10}$ sector read. So we give error probability A a default value of $8/(2.4 * 10^{10})$ (a unit consists of 8 sectors). The rebuild times (which correspondent to 'reconstruction times' in [4]) are read directly from the Figure 8-1 in [4].

4.1 Effect of parity stripe size

MTTDLs as functions of parity stripe size G are plotted in Figure 1. The number of disks (C) in the array is fixed at 21 for all figures. The optimistic estimates for MT_{db} and MT are represented by the solid curves whereas the pessimistic estimates of them by the dot curves. The figure shows that the two values of pessimistic and optimistic estimates of the MTTDLs are close to each other. Both MT_{dd} and MT_{db} decrease as G increases. But the main reasons behind the decreases are different. MT_{dd} is largely affected by the rebuild time (according to formula (2)). As G increases, more surviving disk will participate in the rebuild of the failed disk, which will cause an increase

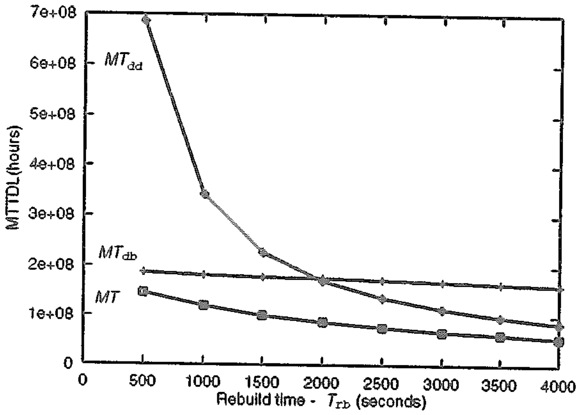

Figure 2: Effect of rebuild time (C=21,G=4)

in rebuild time T_{rb}. As for MT_{db}, it is largely affected by the amount of data on the surviving disks involved in the rebuild of the failed disk. As G increases, more amount of data on the surviving disks will be involved in the rebuild of the failed disk, which will cause an increase in PR_{db}. The figure also shows that when G is small, DD data loss dominates the data loss of the RAID. When G is large, DB data loss dominates the data loss. Based on the above analysis, we can conclude that the MTTDL, MT, depends largely on the parity stripe size (G).

4.2 Effect of rebuild time

Figure 2 depicts the MTTDLs as the functions of the rebuild time. G is 4. Only the pessimistic estimates of MT_{db} and MT are depicted since the optimistic estimates and pessimistic estimates are close to each other both for MT_{db} and MT. It can been seen that with the reduction of rebuild time, there is a little increasement in MT_{db}. DB data loss may occurs after a disk fails due to an UNBE that already exists on the surviving disks when the disk fails (so called an 'old' UNBE), or an UNBE that occurs in failure duration (so called 'new' UNBE). So the former case must dominates in the DB data loss in this situation. As the result, the MT is much less sensitive to rebuild time than the MT_{dd}. A lot of efforts have been made in reducing the rebuild time to improve the reliability of RAID [4][5]. However, it may not be an effective way to increase the MTTDL of the RAID when DB data loss has a relatively big effect on the reliability.

5 Conclusions

We have presented an analytic model to analyse the reliability of declustered-parity RAID focusing on uncorrectable bit errors. We have shown how MTTDL that is calculated by taking into account both DD data loss and DB data loss depends on the parity stripe size and rebuild time by numerical study. MTTDL depends largely on the parity stripe size. MTTDL becomes smaller when parity stripe size is increased. Such MTTDL was less sensitive to rebuild time than the MTTDL that only takes into account DD data loss. Efforts have been made in reducing rebuild time to improve the reliability as only DD data loss is traditionally considered. It may not be so effective when DB data loss is also taken into account.

References

[1] M.Malhotra and K.S.Trivedi, "Data Integrity Analysis of Disk Array systems With Analytic Modeling of Coverage", *Performance Evaluation*, Vol.22, pp. 111-133, 1995.

[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, pp. 145-185, June 1994.

[3] G.A.Gibson, Redundant disk arrays: Reliable, parallel secondary storage, *ACM Distinguished Dissertation Series*, MIT Press, 1992.

[4] M.Holland and G.A.Gibson, "Parity Declustering for Continuous Operation in Redundant Disk Arrays", *Proc. ASPLOS V-10/92/MA,USA*, pp.23-35, 1992.

[5] R.R.Muntz and J.C.S.Lui, "Performance Analysis of Disk Arrays Under Failure", *Proc.16th VLDB Conference*, pp.162-173, 1990.

[6] M.Schulze, G.A.Gibson, R.Katz, and D.A.Patterson, "How Reliable Is A RAID", *Proc. IEEE COMPCON-89*, pp.118-123, 1989.

[7] 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.