Dual-Crosshatch Disk Array: A highly reliable hybrid-RAID architecture

Sunil K. Mishra, Sudheer K. Vemulapalli and Prasant Mohapatra
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011
Email: prasant@iastate.edu

Abstract –*In this paper, we propose a highly reliable RAID architecture called a Dual-Crosshatch Disk Array. It uses the proposed interleaved 2d-parity scheme, a low overhead triple-erasure correcting parity organization. It is a hybrid approach of RAID-4 and RAID-5 with one dedicated parity group and another parity group using block interleaved data and parity. The results obtained from simulations indicate that this architecture possesses extremely high reliability with low overheads, good degraded performance, and acceptable normal-mode performance.*

1 Introduction

To keep pace with the rapidly increasing processing power, disk array architectures have been proposed that can achieve high I/O capacity and performance. Disk array storage systems, such as Redundant Array of Inexpensive Disks (RAID) level-5 data organization [1], provide fault tolerance against disk drive failures, and are cost-effective and have good run-time performance. However, a storage subsystem consists of more than just disk drives. There are controllers for interfacing with the disk drives, cabling to provide data/control paths, power supplies, etc. For the disk array to be fault tolerant it must be able to tolerate failures in any of these components.

There are several known architectures for building disk array subsystems that are tolerant to failures in support hardware components and the disk drives [2]. These architectures employ parity protection, as in RAID-5, with block interleaved data and distributed parity, and are single-erasure tolerant. Such arrays have acceptable mean time to data loss (MTTDL) when the number of disks in the subsystem is small. However, the average number of disks in an installation is growing because of decreasing form factors and increase in the new forms of data needing massive storage capacity like audio and video for multimedia applications. It is projected that by year 2000, average commercial installation will need about 10 TB or more storage capacity [5]. To meet such large storage demand, average installations will need about 5,000 to 50,000 disks. Traditional arrays which can protect from concurrent failure of no more than one disk per parity group will have inadequate reliability for such large storage requirement.

In this paper we present a highly reliable and robust disk array architecture, the *Dual-Crosshatch Disk Array* (DCDA), that is capable of tolerating any three disk failures with minimum number of redundant disks, and any five controller failures. The DCDA uses a novel and efficient parity scheme, the *interleaved 2d-parity*, a variant of the 2d-parity scheme [3]. It is a hybrid approach of RAID-4 and RAID-5 in the sense that one of the parity groups uses block interleaved data and stripped parity while the other uses dedicated parity disks, and hence the name *hybrid-RAID* architecture. The DCDA architecture has extremely high reliability with low check disk overhead, faster data recovery, good degraded-mode performance, and acceptable normal-mode performance making it an attractive solution for designing large disk arrays. The simulation results indicate that the DCDA architecture is order of magnitude more reliable than the crosshatch disk array which is shown to have higher availability than all the other existing disk array organizations. Furthermore, the MTTDL of DCDA is about 10^4 times more than that of the crosshatch disk array. The difference will be even more for the typical values of failure and repair rates.

The rest of the paper is organized as follows. In section 2, we review the advantages and disadvantages of the existing disk array architectures/organizations. In section 3 we describe the DCDA architecture. In section 4, we present the results followed by the conclusions in section 5.

2 Existing Disk Array Organizations

2.1 Single-Erasure Tolerant Architectures

A single-fault tolerant disk array, such as RAID-4 and RAID-5 architectures, provide data accessibility in the presence of any single-failure within the system. In these disk arrays data redundancy is obtained using the parity information for a group of disks, and parity is maintained on a check disk. When a disk fails, the data can be reconstructed by XORing the data on other functional disks from that parity group. The RAID-4 organization uses block interleaved data and dedicated parity. In RAID-5 the parity information is also stripped among all the disks in a parity group along with the data. Stripping of parity results in better write performance [1].

Along with the disk drives, a disk array subsystem must also be tolerant to failures in the support components. Among the existing architectures that tolerate the failure of support hardware, the most prominent ones are Single Path Horizontal Array, Dual Path Vertical array, Dual Path Horizontal Array and Crosshatch Disk Array [2].

In the crosshatch disk array, the disk drives are dual ported and each disk drive is a member of two strings - a horizontal string and a vertical string, and the combination is unique for each drive. Crosshatch refers to the pathing topology of the array architecture. This architecture is tolerant to a single disk

failure per parity group. As each drive can be accessed through two independent controllers, a single controller failure will not affect the data availability from any disk. Any double controller failure can render at most one disk inaccessible. Therefore, after the controllers are repaired, at most one disk needs to be rebuilt. Hence the repair time is less resulting in greater MTTDL.

2.2 Multiple-Erasure Correcting Codes

In order to maintain data integrity under more than one disk failure, more that one redundant disks are required. According to the coding theory, C concurrent disk failures can be tolerated using at least C redundant check disks [3]. Gibson et al. have presented the 1d-parity, 2d-parity, additive-3, and in general, the multidimensional parity scheme [3]. The 2d-parity, a double-erasure correcting code, can tolerate all sets of 3-erasures except the *bad 3-erasures*, and additive-3 code can correct all sets of 4-erasures except *bad 4-erasures* [3]. EVENODD encoding scheme, proposed by Blaum et al. [4], uses a special encoding scheme to store parity information, and can tolerate any 2 disk failures. Burkhard et al. have proposed maximum distance separable (MDS) codes capable of tolerating two or three concurrent disk failures using as many check disks [5].

3 Dual-Crosshatch Disk Array

Most of the existing architectures can tolerate only one disk drive failure per parity group. With the increase in the average number of disks in an installation, these traditional disk arrays may prove to be unreliable. In this section we describe the *Dual-Crosshatch Disk Array* architecture which can tolerate more concurrent disk drive and controller failures than any of the existing architectures.

3.1 The Interleaved 2d-Parity

In case of 2d-parity, parity is computed both along the rows and columns, and is stored in a check disk at the end of each row and column as shown in Fig. 1.a. In the proposed *interleaved 2d-parity* organization, the horizontal parity groups use block interleaved data and parity, unlike in the 2d-parity. The vertical parity groups use dedicated parity disks as in 2d-parity. As a result of parity stripping in the horizontal parity groups an extra check disk is required, compared to 2d-parity, for storing the vertical parity information. Fig. 1.b shows the interleaved 2d-parity scheme. A novelty of this parity scheme is that it can tolerate more number of concurrent failures than the number of redundant check disks required per parity group.

3.1.1 Failure Recovery – Under a single disk failure, data recovery is possible using either horizontal or the vertical parity group. For any double disk failure in any single parity group data can be recovered using the two orthogonal parity groups associated with the failed disks. However, when three disks fail, say disk *5*, *P1*, and *9* as in Fig. 1.b, the data for disk *5* cannot be recovered using any single parity group. Data pertaining to disk *P1* and *9* can be reconstructed from the associated vertical and horizontal parity groups respectively; then the data on disk *5* can be reconstructed. If each parity group contains G data disks and a check disk, a total of $3G-1$ disk accesses is required to construct the data on the three failed disks. Hence, the interleaved 2d-parity scheme is 3-erasure correcting.

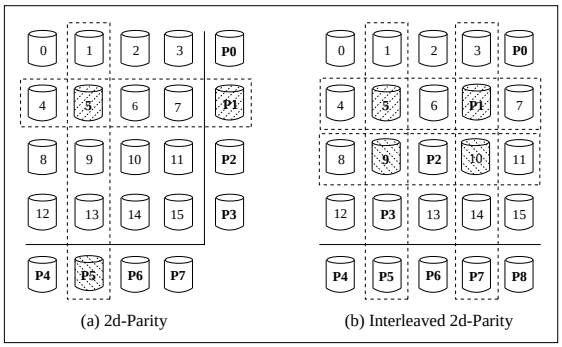

Figure 1: 2d-parity and Interleaved 2d-parity schemes

The interleaved 2d-parity scheme can also tolerate all sets of 4-erasures except the *bad 4-erasure* as shown by shaded disks in Fig.1.b. The bad 4-erasures are unrecoverable as two disks fail in each of the four parity groups. With this parity organization, there is no chance of data loss for any number of disk failures until a bad 4-erasure occurs. For an NxN disk array using interleaved 2d-parity, a maximum of $2N-1$ disks can fail without the occurrence of a bad 4-erasure. The probability of occurrence of a bad 4-erasure is very low, and hence this scheme provides very large MTTDL.

3.1.2 Overhead – The *check disk overhead* is defined as the ratio of the number of check disks to information disks. For c check bits, the interleaved 2d-parity code can have a maximum of $(c-1)^2/4$ information bits as compared to the $c^2/4$ information bits for the 2d-parity code. In a disk array having G information disks per parity group and n such parity groups (we assume $n \leq G$), total of $n+G+1$ check disks are required for the interleaved 2d-parity, whereas the 2d-parity needs $n+G$ check disks. When n is equal to G, the number of check disks required is $2G+1$. The check disk overhead for this scheme is $(2G+1)/G^2$ as compared to $2/G$ in case of 2d-parity. As G is more than 1, less than 3 redundant check disks are required (per data group) for this 3-erasure correcting code. The proposed parity organization is thus optimal in terms of check disk overhead.

3.1.3 Update Penalty – The *update penalty* [3] for a disk array is the number of check disk updates required for any write to an information disk. The minimum update penalty for any t-erasure correcting code is t [3]. For the proposed parity organization, any write to a disk needs updating the parity information in both the parity disks (in case of horizontal parity group, this is the disk containing parity for that block of data) and the dedicated parity disk corresponding to the vertical parity group. With reference to Fig.1.b, a write to disk *5* involves updating parity information in disk *P1, P5 and P7*. Thus the proposed scheme has the update penalty of 3, and this is optimal for a triple-erasure-correcting code. Several schemes have been proposed to improve the write performance of RAID-5 [6], and can also be employed in this scheme to improve the performance. The performance issues are discussed in Section 3.3.

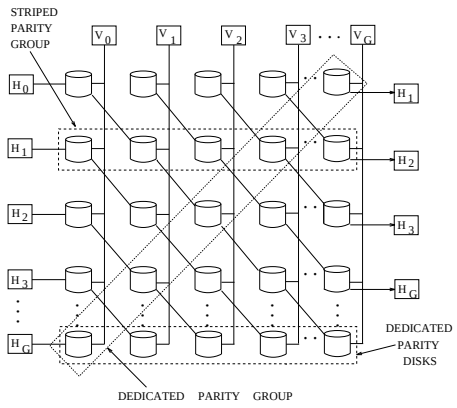

Figure 2: Dual-Crosshatch Disk Array.

3.2 The DCDA Architecture

Dual-Crosshatch Disk Array can be formally defined as an array in which the disk drives are dual ported and each disk is a member of two controller strings and two parity groups, and the combination is unique for each disk drive as shown in Fig. 2. The two controller strings are – (a) horizontal controller string and (b) vertical controller string. The two parity groups are – (a) horizontal-stripped parity group and (b) diagonal-dedicated parity group. The DCDA architecture uses the interleaved 2d-parity scheme, but instead of a vertical parity group we chose to use a diagonal parity to insure better performance. The DCDA can tolerate any three disk failures and all sets of 4-disks failures except the bad 4-erasures. As stated before, *crosshatch* refers to the pathing topology of the architecture. This architecture is referred to as *dual-crosshatch* because of the presence of additional diagonal-dedicated parity groups.

A single controller failure does not affect data availability from any disk. Under a vertical controller failure, access to at most one disk in each of the horizontal parity groups is affected. When a horizontal controller fails, disk access to at most two disks in a diagonal parity groups is affected. However, if we use a vertical parity group instead of diagonal parity group, in the case of vertical controller failure, access to all disks in that parity group is affected. All the parity disks of the diagonal parity groups are located in the last row of the disk array. This has the advantage that each of these disks are connected to two independent controllers. Since, parity information needs to be updated in these disks for a write to any data disk, two sets of controllers provide some degree of redundancy in the controller path. Any two controller failures renders at most one disk inaccessible in both the DCDA and the crosshatch diagonal disk array. Hence, after the controllers are repaired, only one disk needs to be rebuilt.

The DCDA architecture can tolerate any five controller failures as compared to triple controller failure tolerance of the crosshatch diagonal disk array. It can tolerate most of the six controller failures except for a few cases which cause bad 4-erasures. Hence, this architecture essentially eliminates the controller failure as a contributor to the unavailability of the storage subsystem.

Assuming that G^2 information disks are organized in a square array of side G, then the DCDA organization needs $2G+1$ redundant disks as compared to $2G$ redundant disks for EVENODD and the 2-erasure correcting MDS codes. For example, for a total of 1024 disks organized in a 32 x 32 array, 63 check disks would be required; so the check disk overhead is about 6.5%. The check disk overhead for the DCDA is nearly same as that of the EVENODD encoding scheme for large disk arrays, but the former can tolerate triple-erasures as compared to the double-erasure tolerance of the EVENODD parity scheme.

Considering a single disk failure with all the controllers functional – one set of controllers can access the functional disks of the horizontal parity group and the other set of controllers can access the functional disks of the vertical parity group. By using both the parity groups to reconstruct different blocks of data in parallel, data reconstruction time can be reduced. A shorter repair time decreases the probability of second disk failures and increases the MTTDL of the system.

The major advantage of the DCDA is that it is tolerant to more concurrent controller and disk drive failures than the other prior-art architectures. For example DCDA can tolerate any three disks and any one controller failures, or any disk and any five controller failures. Hence the DCDA is a better fault tolerant system as a whole than the existing architectures.

3.3 Performance Issues of DCDA

The performance of DCDA is comparable to that of crosshatch disk array except for the cases of small write and small read-modify-writes (RMW). Under normal-working-mode, a read request for a large chunk of data distributed over two or more parity groups can be handled in parallel using the two set of controller strings. Similarly, two different read requests for data spread over different parity groups can be handled in parallel. So, the read performance of the DCDA is nearly twice that of the RAID organizations.

Since DCDA uses the interleaved 2d-parity scheme, the update penalty for a write operation is 3. A small write needs four RMW accesses to the disk containing data and the three related parity disks. If both the controller strings are free, they can be used to write to the horizontal parity group and the dedicated parity disks in parallel. For example, a write to disk 5 in Fig. 1.b needs updating parity information on disk P1, P5, and P7. In the DCDA architecture, the horizontal controller string can be used to write to disk 5 and P1, while the other set of controllers can write to disks P5 and P7. For large write requests use of both the controller string ensures nearly same performance as RAID-5. So this high update penalty is partially nullified by the extra set of controllers. But when a controller string is busy, or under a few controller failures, the above assumption does not hold good, and small write and RMW performance may not be acceptable. We suggest the use of data cache in the controllers to improve the write and RMW performance. A number of schemes have been proposed for performance improvement using cache memory and other ways [6]. This architecture also has better degraded-mode performance characteristics as data on the failed disks can be reconstructed faster and made available. Hence this architecture has better throughput, in degraded mode, as compared to the other existing architectures.

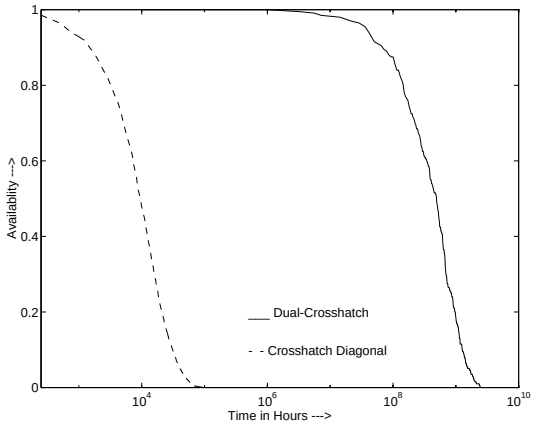

Figure 3: Availability vs Time for a 1024 disk array.

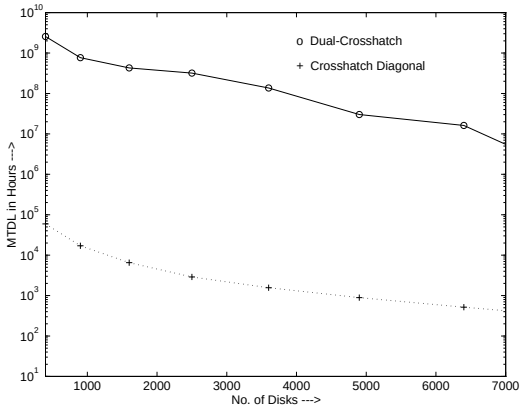

Figure 4: MTTDL vs Storage Capacity.

4 Results

The results presented in this section are based on event driven simulations of the different disk array architectures. We assume that the mean time to failure of the individual disks and controllers are independent and exponentially distributed. Typical value for MTTF of a disk drive is in the range of 50,000 to 400,000 hours and the corresponding MTTR with a hot spare is 1 to 2 hours. The corresponding values for the controllers (including the cable link) are 100,000 hours and 20 hours. With the typical values, the MTTDL of the DCDA is extremely high making it difficult to run the simulations for such large values of time. So we have considered the following values – MTTF and MTTR for disks to be 50,000 hours and 5 hours respectively, and those values for the controllers are 50,000 hours and 20 hours respectively. As we are comparing the availability of the two disk array architectures using these values gives an idea of the relative improvements.

In Fig. 3, we show the availability of the system with respect to time. A disk array consisting of 1024 disks arranged in a 32 x 32 array is considered here. As can be seen from the plot the availability of the DCDA architecture is very high compared to the crosshatch diagonal disk array. This is because the crosshatch disk array uses a single-erasure correcting code whereas the DCDA uses a triple-erasure correcting code. Furthermore, the occurance of the bad 4-erasure cases that result in data loss is very rare.

Fig. 4 shows the variation of MTTDL with storage capacity of both crosshatch diagonal and DCDA architectures. All the failure and repair rates used are same as those for Fig. 3. As can be seen from the plot, the reliability of both architectures decreases as the disk arrays are scaled up in dimension. But the MTTDL of the DCDA is more than that of the crosshatch diagonal disk array by a factor of around 45,000 for a disk array consisting of 1024 disks with the used parameters. This factor is even larger for the typical values of failure and repair rates as mentioned earlier in this section.

5 Conclusions

In this paper, the *Dual-Crosshatch Disk Array*, a new architectural alternative for configuring a redundant disk array is presented. Compared to the prior known array architectures, the DCDA provides higher fault tolerance, higher availability, better degraded-mode performance, and faster reconstruction. Since the DCDA uses the interleaved 2d-parity encoding scheme, the check disk overhead is almost same as that of the EVENODD and other double-erasure correcting schemes, but the DCDA can tolerate any triple-erasure. The array is fault tolerant to any five controller failures. Hence the DCDA is a robust and highly reliable storage subsystem.

References

[1] D.A. Patterson, G.A. Gibson, and R.H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *ACM SIGMOD Intl. Conf. on Data Management*, pp. 109-116, 1988.

[2] S.W. Ng, "Crosshatch Disk Array for Improved Reliability and Performance," *Intl. Symposium on Computer Architecture*, pp. 255-264, 1994.

[3] Gibson et al., "Failure Correction Techniques for Large Disk Arrays," *Third Intl. Conf. on Architectural Support for Programming Language and Operating Systems*, pp. 123-132, April 1989.

[4] M. Blaum et al., "EVENODD: An Optimal Scheme for Tolerating Double Disk Failures in RAID Architectures," *Annual Intl. Symposium on Computer Architecture*, pp. 245-254, 1994.

[5] W.A. Burkhard, and J. Menon, "Disk Array Storage System Reliability," *Annual Intl. Symposium on Fault Tolerant Computing*, pp. 432-441, 1993.

[6] P.M. Chen et al., "RAID: High-Performance, Reliable Secondary Storage," *ACM Computing Surveys*, Vol. 26, No. 2, pp. 145-185, June 1994.