available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

RAID-RMS: A fault tolerant stripped mirroring RAID architecture for distributed systems

Javad Akbari Torkestani[a,*], *Mohammad Reza Meybodi*[b,c]

[a]*Computer Engineering Department, Islamic Azad University, Arak, Iran*
[b]*Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran*
[c]*Institute for Studies an Theoretical Physics and Mathematics (IPM), School of Computer Science, Tehran, Iran*

ARTICLE INFO

Article history:
Received 5 September 2006
Received in revised form
14 June 2008
Accepted 2 September 2008

Keywords:
RAID
Disk mirroring
Reliability
Data striping

ABSTRACT

Disk arrays, or RAIDs, have become the solution to increase the capacity, bandwidth and reliability of most storage systems. In spite of its high redundancy level, disk mirroring is a popular RAID paradigm, because replicating data also doubles the bandwidth available for processing read requests, improves the reliability and achieves fault tolerance. In this paper, we present a new RAID architecture called RAID-RMS in which a special hybrid mechanism is used to map the data blocks to the cluster. The main idea behind the proposed algorithm is to combine the data block striping and disk mirroring technique with a data block rotation. The resulting architecture improves the parallelism reliability and efficiency of the RAID array. We show that the proposed architecture is able to serve many more disk requests compared to the other mirroring-based architectures. We also argue that a more balanced disk load is attained by the given architecture, especially when there are some disk failures.

1. Introduction

A Redundant Array of Independent Disks-RAID system is composed of a disk array, and a disk array controller. The controller's function is to receive an operation from the user of the disk array, and to carry it out by performing a set of low-level operations on specific disks. There are several RAID architectures that are classified into five levels (Patterson et al., 1988; Gibson et al., 1998). Different RAID architectures can be distinguished based on their type of encoding, mapping and algorithms used to access data. In other words, each RAID system can be defined by these three components. The RAID architecture proposed in this paper carefully considers all the subjects above. The encoding indicates the type of redundancy information used to encode the data. The mapping introduces the pattern used to place data and redundant information on the disk array. Algorithms used to access data can be classified as normal mode and failed mode ones. In normal mode, either there is no failure in the disk array or the controller knows about the failed disks, if any. In failed mode, a disk failure has occurred in the middle of controller operation. The controller then needs to recover the error and complete the operation. This process is called error recovery.

There have been numerous reliability and performance evaluation studies of RAID, since the publication of Patterson et al. (1988). This paper focuses on the reliability and bandwidth, which are the main reasons of formation RAID systems. Gibson et al. (1998), Thomasian and Blaum (2006) and Thomasian and Blaum (2006), Thomasian (2006) presented a reliability-based comparison among some existing RAID architectures, especially for those are based on mirroring technique. There are several studies considering new disk mirroring organizations in which the increasing disk loads due to the failures are balanced. For example, interleaved

* *Corresponding author*. Computer Engineering Department, Islamic Azad University, Arak, Iran Tel.: +98 8613663041; fax: +98 8613663042.
E-mail addresses: j-akbari@iau-arak.ac.ir (J. Akbari Torkestani), mmeybodi@aut.ac.ir (M. Reza Meybodi).
0167-4048/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2008.09.001

declustering method-ID (Tera data Corporation, 1985), implemented in the original teradata database machine, partitions N disks into c clusters and the primary data on each disk is distributed evenly among the other disks in the cluster. Hsiao and DeWitt (1993) introduced a RAID architecture called chained declustering which is an improvement over ID and tolerates a larger number of disk failures than ID. In this architecture, each disk is partitioned into a primary and secondary area, and the primary data on each disk is replicated on the secondary area. Chen and Towsley (1996) proposed the group rotate declustering technique in which similar to the basic mirroring, disks are divided into primary and secondary disks. Data is striped and the stripe units of each primary disk are located in a rotation manner on secondary disks.

The RAID architectures, based on their redundant data, provide different reliability levels and effective bandwidth to access data. In spite of the high redundancy level in RAID-1, due to the basic mirroring technique, the architecture cannot balance the load increased by disk failures and so it reduces the reliability and access bandwidth for processing read requests. To overcome this shortcoming, we propose a new RAID architecture in which the data block striping and disk mirroring techniques are combined. The former technique is used to balance disk loads and improves the bandwidth to access data and the latter technique replicates the data blocks and increases the reliability of the RAID array. The theoretical analysis shows that the proposed architecture outperforms the existing mirroring-based RAID architectures in investigated metrics and the experimental results completely confirm it. It is shown that the proposed architecture is able to serve many more disk requests compared to the other RAID systems whose architectures are based on disk mirroring scheme. We also argue that a more balanced disk loads is attained by the given model, especially when there are some disk failures.

The rest of the paper is organized as follows. In the next section, we propose a new RAID architecture, which is explained in full detail in Section 3. The proposed architecture is evaluated through theoretical analysis in terms of the maximum bandwidth, disk operation cost and reliability in Section 4, and then compared to other similar architectures through simulation in Section 5. The concluding remarks are presented in Section 6.

2. The proposed RAID architecture

In this paper, we adopt the following notational conventions to introduce the proposed RAID architecture. Let N and n denote the block sequence number in the memory address space and the number of disks in the cluster, respectively. M denotes the number of blocks per disk, each of size S and i and j are indices to specify the horizontal and vertical positions of a data or parity blocks, respectively. Formally, a redundant array of independent disks-RAID can be described by a quintuple $< \tilde{A}, \underline{B}, \underline{R}, f_B, g_R >$, where $\tilde{A} = \{a_0, a_1, \ldots, a_N\}$ denotes the sequence of the data blocks, $\underline{B} = \{b_{(i,j)} | 0 \le i \le m-1, 0 \le j \le n-1\}$ is the finite set of data blocks in the disk array, $\underline{R} = \Phi \cup \{r_{(i,j)} | 0 \le i \le m-1, 0 \le j \le n-1s\}$ is the finite set of redundant blocks which may be a null set. It implies that an architecture may provide no redundancy, like RAID0. Let x_B and $x_R$be the cardinality of $\underline{B}$ and $\underline{R}$, respectively. The redundancy rate for a given RAID architecture can be obtained by $x_R/(x_B + x_R)$. f_B:A → B is the block function, which maps to each block sequence number a_N a two-dimenssional address $b_{(i,j)}$ in the cluster. Let b denotes a given subset of $\underline{B}$. g_R:B → R is the redundant function, which assigns a redundant block $r_{(i,j)}$to a given subset b. It gives rise to the RAID architectures can be easily studied.

The basic idea behind the new hybrid RAID architecture called RAID-RMS[1] is to combine the data block striping and disk mirroring techniques with the block rotation to improve the access bandwidth and reliability of the RAID array. To do so, the data blocks are divided into two distinct set of blocks: The primary data blocks and the Mirrored (or Backup) blocks. Each disk is also partitioned into a primary and secondary area (Hsiao and DeWitt, 1993) and the former data blocks are distributed in the primary disk area and the latter ones in the secondary. In this RAID architecture, Data is striped (Chen et al., 1994) and the primary data is distributed evenly among all the n disks in the cluster, then the resulting primary data blocks on each disk are replicated among the other $(n-1)$ disks in a rotated manner fully described in the next section. Now, if a single disk fails, an effectively balanced read load can be attained at all surviving disks. So, this architecture can increase the parallelism for processing read requests, the reliability and the effective bandwidth to access data.

It is shown that the proposed architecture is able to serve many more disk requests compared to the other RAID systems whose architectures are based on disk mirroring. Besides, such architecture can tolerate more disk failures and decrease the rate of missing data due to the presented data distribution. The disk failure may cause a negligible reduction to access data and system performance compared to the other existing architectures (Akbari Torkestani and Meybodi, 2005a; Akbari Torkestani and Meybodi, 2005b; Akbari Torkestani et al., 2006; Akbari Torkestani and Haghighat, 2003; Cortes, 1999; Stonebraker et al., 2000).

The method proposed to distribute the data, forms the cluster so that each disk includes both primary and backup data copies. That is, the primary data blocks are located in the upper halves of the disks (i.e. primary area) and their mirrored copies are located in the lower halves (i.e. primary area). The most important issue which should be considered is designing an efficient data distribution algorithm for distributing the blocks in the manner mentioned earlier. Such method description is comprehensively presented in the next section.

In this data distribution method the data blocks and their backups are distributed by a right rotation in the separated disks. Thus, less data will be lost after disk failures and it can be used as a data storing model for those systems in which data correction, protection and security are really important. Furthermore, in this model, while the system is in the recovery state to repair some failed disks or blocks, other blocks and disks remain available and this improves the system performance dramatically.

[1] Rotating-Mirroring-Striping (RMS).

3. Description of RAID-RMS

As mentioned earlier, the proposed architecture suggests a combination of the data block striping and disk mirroring techniques with a data block rotation to distribute the data in the cluster. In the remaining of this section, the data distribution method proposed in this paper will be formulated and described in detail. The formulas as shown by (1)–(3) provide two mappings from the block sequence numbers to the primary data blocks in the cluster and vice versa, as well as a mapping from the data blocks to their backup copies. In other words, the block function f_B:$A \rightarrow B$ and redundant function g_R:$B \rightarrow R$ can be realized by the mentioned formulas. Since the sequence block numbers in the memory space are addressed in a single-dimension fashion, they must be transformed to the two-dimension addresses to be stored in the cluster.

In our proposed RAID array, each disk is partitioned into a primary and secondary area (Hsiao and DeWitt, 1993). The data blocks are also divided into two distinct set of blocks: The primary data blocks and the mirrored blocks. Let $B_{(i,j)}$'s denote the primary data blocks which are located within the upper halves of the disks, where $0 \le i \le n-1$and $0 \le j \le m/2-1$, and$M_{(i',j')}$'s denote the mirrored copies located in the lower halves, where $m/2 \le i' \le m-1$ and $0 \le j' \le n-1$. Now, the data distribution method can be described as follows: when the RAID controller receives a request to store a single or a set of data blocks, the following steps should be taken by the controller:

Step 1. Data is striped (Chen et al., 1994) and the primary data is distributed evenly among all the n disks in the cluster. The addresses in which the primary data blocks should be located are obtained by (1). This formula defines the block function f_B:$A \rightarrow B$ for our method, where B denotes the primary data block. It's a mapping from the block sequence number A_N to the physical address (primary data block) of the cluster $B_{(i,j)}$.

$$B_{(i,j)} = \begin{cases} i = \lfloor \frac{N}{n} \rfloor \\ j = \left[\lfloor \frac{N}{n} \rfloor + (N \bmod n) \right] \bmod n \end{cases} \quad (1)$$

Where $0 \le i \le m/2-1$ and $0 \le j \le n-1$. Let $A_N = N$ for notational convenience.

Step 2. In this step, the primary data blocks on each disk are replicated in a rotated manner among the other $(n-1)$ disks modula-n as stated by (2-a). It maps a mirrored block $M_{(i',j')}$ to each block sequence number A_N and defines the redundant function g_R:$B \rightarrow R$.

$$M_{(i',j')} = \begin{cases} i' = \lfloor \frac{N}{n} \rfloor + m/2 \\ j' = \left[\lfloor \frac{(N+1)}{n} \rfloor + ((N+1) \bmod n) \right] \bmod n \end{cases} \quad (2\text{-a})$$

Where $m/2 \le i' \le m-1$ and $0 \le j' \le n-1$. In the steps above, a right side rotation technique is implicitly involved to find the position of data and mirrored blocks within disk array. For retrieving the missing blocks, the data blocks and their backups need to be immediately converted, so (2-a) should be rewritten based on i and j. Substituting $\lfloor N/n \rfloor = i$ and $j' = j+1$ in (2-a) finally yields (2-b).

$$M_{(i',j')} = \begin{cases} i' = i + m/2 \\ j' = (j+1) \bmod n \end{cases} \quad (2\text{-b})$$

When disk i fails, the corresponding backup copies of the missing primary data blocks, $B_{(i,j)}$'s(for all $j = 0,1,\ldots,m/2-1$), can be directly found by (2-b) and easily retrieved. The lost mirrored blocks $M_{(i',j')}$ (for all $j = m/2,\ldots,m-1$) can be reconstructed by inversely applying (2-b). Besides, we represent an inverse-block function in which the Cartesian coordinates $B_{(i,j)}$ of the block-interleaved data are mapped to the sequence block number A_N.

$$A_N = \begin{cases} (i \times n) + (j \bmod n) - i; & i \le j \\ (i+1) \times n + j - i; & i > j \end{cases} \quad (3)$$

Where $0 \le i \le m/2-1$and $0 \le j \le n-1$.

4. Evaluation and comparison

As mentioned earlier, our proposed RAID architecture exploits the data striping technique in order to balance the disk loads and the disk mirroring technique to retrieve the missing blocks. Therefore, it seems appropriate to implement in those storage systems which hold huge amounts of critical data. In the remaining part of this section, the presented architecture is theoretically analyzed in terms of the maximum bandwidth, disk operation cost and reliability and its results are compared to those of the existing mirroring-based RAID architectures.

4.1. *The maximum bandwidth*

The RAID architectures, based on their redundant data, provide different reliability levels and effective bandwidth to access data. These are some criteria for evaluating the effectiveness of the disk storage architectures. The access bandwidth needs to be considered when the RAID system is either in the normal or failed modes to process the requests. Although the proposed architecture, due to the block striping technique, is able to access the total bandwidth provided by all disks for processing both small and large read requests, the mirroring-based architectures and, consequently, the proposed architecture replicate the data blocks on the secondary disks so the data blocks and their backups would be involved for processing both small and large write requests. Therefore, all the RAID architectures whose coding methods are based on the data mirroring technique, can efficiently exploit only half of the total bandwidth available to process large writings.

At first, we introduce some concepts upon which the next discussions are based in the rest of the paper. Small Read/Write requests are those kinds of disk operations in which the whole or part of a data block is processed and their scopes do not exceed the given block at all. Therefore, such I/O operations may influence the different blocks of the same or different stripes simultaneously (Corbett et al., 1993). Large Read/Write requests are those kinds of disk operations in which the several blocks of the same stripe or even the different stripes may be involved to process the requests. Note that, in most cases, the empirical results given in the literatures are not as good as the theoretical results are. This is because the bandwidth provided by the architecture is considered in a maximum case and so it is not always reachable (Cormen and Kotz, 1993; Foster et al., 1997; Ho et al., 2000).

Let n and B be the total number of disks in the arrays and the maximum disk bandwidth, respectively. As shown in Table 1, the maximum bandwidth provided by all given disk arrays to process the small read requests is $n \times B$. In spite of the high redundancy level in RAID-1, due to the disk mirroring technique, the effective bandwidth provided by the model is limited to only half of the bandwidth available in the cluster. So when a large read is requested to process, the RAID controller is able to process only two consecutive data blocks in parallel. The maximum bandwidth provided by the other existing block-interleaved architectures is $n \times B$. In RAID-5, to solve the bottleneck problem which is associated with modifying the parity blocks in RAID4, the individual parity disk is removed and the parity blocks are distributed across the all disks. Since the given parity blocks should be modified after processing the small or the large write requests, the maximum bandwidth available for processing small write operations is reduced to $(n-1)/2 \times B$ and for large write operations to $(n-1) \times B$.

4.2. Disk operation cost

Let $N_{\text{Totaldisk}}$ and $N_{\text{Checkdisk}}$ be the total number of disks and the number of check disks in the cluster, respectively. Let T_R be the mean time to read a given block and L_{Data} be the length of a given request (i.e. the number of blocks per file). As mentioned before, in all RAID architectures discussed in the previous sections, n individual small read requests can be processed in parallel, where $n = N_{\text{Totaldisk}}$. As shown in Table 1, the average cost, due to processing the small read operations, using such RAID architectures is T_R. We introduce the following formula to calculate the disk operation cost needed to process a Large read request as long as L_{Data}, where L_{Data} is the number of blocks needed to be processed. This formula can be applied for all RAID architectures except RAID-1. The resulting costs have been shown in Table 1.

$$(L_{\text{Data}} \times T_R)/(N_{\text{Totaldisk}} - N_{\text{Checkdisk}}) \quad (4)$$

The proposed architecture not only does not cause any side effects on seek time and disk cache hit, but it achieves better results than the other disk arrays. In RAID-5, the parity blocks, associated to the modified strips, should be updated after each small or large write operation. This process is described as follows: the modified data blocks and the old parity block should be read to obtain the new parity block. Then the new one is rewritten. Thus, as shown in the last two columns of Table 1, each write request has an additional cost for reading the modified data and the old parity blocks.

4.3. Reliability analysis

The more parallelism between requests and the higher bandwidth to access data, due to the striped data blocks, and the higher reliability against disk failures due to the data mirroring technique, are some of the advantages of the proposed RAID architecture. In this subsection, the reliability of the proposed RAID architecture is evaluated in terms of the maximum disk failure coverage, which means the maximum number of disk failures that can be tolerated by the disk array (Cortes, 1999; Anderson et al., 1996), and then compared to those of the arrays mentioned earlier.

As mentioned before, due to using the data mirroring technique in RAID-1 and RAID-10 to backup the disk failures, the levels of the failure recovery provided by both given RAID architectures are the same. Assume that two independent disks fail successively so that the second failure occurs before the first one can be recovered completely. It implies that the rebuild process which is retrieving the first failed disk cannot be completed before the second disk failure. Since the MTTR[2] is assumed to be much smaller than the MTTF[3], the probability that the failed disk can be retrieved and the RAID array returned to the original state is close to unity. Therefore, the probability that each disk failure or missing data block can be recovered by the given architectures is $1 - (1/n(n-1))$, where n is the total number of disks in the cluster. Despite their high redundancy level, because of the weak data distribution pattern, these arrays cannot tolerate the disks failures effectively. But, there is a special case, when these architectures tolerate $n/2$ successive independent disk failures (Cormen and Kotz, 1993; Foster et al., 1997; Harry et al., 1995; Howard et al., 1988; Hsiao and DeWitt, 1990; Hu et al., 1999).

RAID-5 uses the distributed parity blocks to detect and correct errors. In this architecture, the controller checks the strips by xoring the data blocks and parity block, and recognizes the happened failure, if any. RAID-5 can only tolerate the single disk failures and detect the strips in which some failures happened and it is not capable of exactly detecting the damaged data blocks. In the case that, the number of failures happened in a given strip is even, the controller of such a RAID array cannot detect any failures (Asami et al., 1999; Cabrera and Long, 1991; Cao et al., 1994).

As mentioned earlier, RAID-RMS uses a special hybrid mechanism to map the data blocks to the cluster. The main idea behind the proposed algorithm is to combine the data block striping and disk mirroring techniques with a data block rotation to improve the parallelism, reliability and efficiency of the RAID array. In this architecture, the data is block-interleaved and then distributed in the other $(n-1)$ disks modula-n. The mirrored blocks are also replicated diagonally throughout the secondary area in a similar manner. Such a RAID architecture, whose data and mirrored blocks are distributed diagonally, alleviates the effects of the both disk and block failures. The results of the comparison between these architectures show that the proposed architecture leads to considerably better results of both data block and disk failure recoveries. Assume that two successive independent disk failures be happened in the different disks of the cluster so that the second failure occurs before recovering the first one completely. In this case, the probability that the missing data blocks can be recovered is $1 - (1/n(n-1)(n-2))$.

The RAID system we considered in Section 4.1 to evaluate and compare the given RAID architectures, concerned the maximum bandwidth provided by the models when the system is in the normal state. In this state, the RAID controller does not need to perform any special functions to support the RAID system. In the remaining part of this section, we consider the maximum available bandwidth of the RAID architectures when the RAID system is in failed mode. In this

[2] Mean time to repair.
[3] Mean time to failure.

Table 1 – The theoretical results of the four RAID architectures in terms of the maximum I/O bandwidth, I/O rate and recovery rate

	RAID level	Small read	Large read	Small write	Large write
Maximum bandwidth	RAID-1	$N \times B$	$B/2$	$(n/2) \times B$	$(n/2) \times B$
	RAID-10	$n \times B$	$n \times B$	$(n/2) \times B$	$(n/2) \times B$
	RAID-5	$n \times B$	$(n-1) \times B$	$(n-1)/2 \times B$	$(n-1) \times B$
	RAID-RMS	$n \times B$	$n \times B$	$(n/2) \times B$	$(n/2) \times B$
Read/write cost	RAID-1	R	$L/2 \times R$	W	$L \times W$
	RAID-1 0	R	$L/n \times R$	W	$L \times W/(n/2)$
	RAID-5	R	$L \times R/(n-1)$	R + W	$L \times (R+W)/(n-1)$
	RAID-RMS	R	$L/n \times R$	W	$L \times W/(n/2)$
Recovery rate	RAID-1	$(n-1) \times B$	B		
	RAID-10	$(n-1) \times B$	$(n/2) \times B$		
	RAID-5	$(n-1) \times B$	$(n-2) \times B$		
	RAID-RMS	$(n-1) \times B$	$(n-1) \times B$		

state, the RAID controller is recovering the disk failures. First of all, the following questions need to be considered.

1 How much is the RAID architecture able to utilize the available bandwidth after some disk failures?
2 How much is the RAID system able to utilize the available bandwidth of the cluster for retrieving the missing data?

To answer these questions, we describe the recovery process in detail. When a disk fails, rebuild processing is used to retrieve the contents of the failed disk on a spare disk. During the reconstruction process, the normal functions of the RAID system should be temporarily stopped. Therefore, to achieve higher performance, the reconstruction interval should be shortened as much as possible. It improves the reliability of the array as the MTTR/MTTF ratio is reduced.

Since all the missing data blocks should be rewritten sequentially, the expected time to rewrite a failed disk is the same for all RAID architectures. So we consider the expected time to do the reading process. In RAID-1 and RAID-10, this process sequentially reads the contents of the mirrored disk and the available bandwidth to reconstruct the failed disk is limited to only the mirrored disk. In RAID-5, this process involves the reading of successive blocks from the $(n-1)$ surviving disks. Although these reading operations can be done in parallel, but the mentioned process should be repeated for all the missing blocks and so here the expected time is also similar to those of the earlier architectures. The proposed architecture can involve the $(n-1)$ surviving disks to read the block-interleaved backup copies simultaneously and so provides the highest effective bandwidth compared to the other architectures. As shown in Table 1, after a single disk failure, the proposed RAID model provides the most effective bandwidth compared to the other architectures as we argued in Section 5.1. Furthermore, the proposed RAID array can tolerate all double disk failures whereas the other arrays will lose some data. In Table 1, let $N_{total} = n$, $T_R = R$, $T_W = W$ and $L_{Data} = L$ for notational convenience.

5. Experimental results

We conducted some simulation experiments to study the performance of the proposed RAID architecture, and compared the obtained results to the other studied architectures in terms of the access bandwidth, I/O rate and reliability.

We first experimented the effective bandwidth provided by the studied RAID architectures to access data. The parallel reads and writes are considered separately and the large request size, for either read or write operations, is set to 20 MB. The size of each block strip is set to 32 KB. This configuration implies a high degree of striping and parallelism. The requests to access disks and data are generated with a uniform distribution. The measured access bandwidth has been shown in Table 2 as a function of the number of client requests for parallel data accesses. The results summarized in all tables are averages over one hundred runs. We considered both light and heavy I/O traffic rates, and varied the number of client requests from 1 to 16 parallel client requests, while the number of disks is fixed at 16. To reveal the parallel I/O capability of the disk arrays, the large read and the large write are studied and to test the individual disk performance, the small read or write are considered.

For a small read, the access bandwidth provided by RAID-5 is very close to that of our proposed disk array and RAID-RMS performs only slightly better than RAID-5, while it shows much better results than RAID-10. As the number of client requests increases, both RAID-RMS and RAID-5 show higher scalability and RAID-10 lags behind. The simulation results for large read requests are very close to that for small read requests.

For parallel writes of either a large file or a small block, the RAID-RMS achieves the best scalability among the other

Table 2 – Maximum bandwidth comparison of RAID-RMS, RAID-10 and RAID-5 and the effects of I/O rate (MB/S)

RAID architecture	Number of clients	Small read	Small write	Large read	Large write
RAID-10	1	2.32	2.45	2.27	2.21
	8	6.23	6.02	6.02	6.12
	16	9.85	9.9	10.43	9.68
RAID-5	1	2.40	1.92	2.34	2.01
	8	10.11	4.14	10.07	4.03
	16	15.97	5.15	16.00	6.12
RAID-RMS	1	2.51	2.45	2.45	2.57
	8	10.58	9.47	9.97	10.23
	16	16.13	15.39	16.05	15.98

architectures with a highest 16 MB/s for 16 clients. RAID-5 scales slowly due to the bottleneck problem mentioned earlier for modifying the parity blocks. RAID-10 scales slower than our proposed architecture, but very faster than RAID-5. To sum up, the RAID-RMS, due to the full parallelism in accessing all data blocks in the upper half of the disk array, outperforms the others.

Since more disks in the cluster may satisfy more client requests, scaling of a disk array size may increase the effective access bandwidth. On the other hand, it may also introduce more contentions. The effects of disk array size are very similar to the effects of the client request rate. In both cases, we consider the saturated cases of accessing a large number of client requests simultaneously. These experiments take into account the array size in the simulations and measure the access bandwidth provided by the architectures mentioned above, as the number of disks increases from 2 to 16 in array. The total number of client requests is fixed at 16. The workload is fixed at 320 MB for large reads or for large writes, regardless of the array size. For small reads or writes, the fixed workload is set to 512 KB for 16 clients.

As shown in Table 3, these experimental results are slightly different from those of the earlier ones, especially for read operations. For large or small writes, the ranking given for previous experiments remains unchanged and RAID-RMS is ranked above the other architectures and RAID-5 below. The differences in access bandwidth are attributed mainly to the architectural characteristics of each disk array. The strength of the RAID-RMS lies mainly in its superior performance in performing parallel write operations, regardless of the write granularity.

The I/O rates have been shown in Table 4 against the number of disks in the cluster. In these experiments, the number of client requests is fixed at 16, the number of disks increases from 2 to 16 and each client requests to read or write 500 MB data on the array. The block read results show that the RAID-5 and RAID-RMS perform better than RAID-10. Especially, as the array size increases to 16 disks, the proposed array and RAID-5 converge to an output rate of around 4 MB/s. But the RAID-10 speed lags far behind because it experienced more overheads. This overhead comes from a less aggressive mirroring scheme on the RAID-10. Table 4 also shows the results of the block write experiments, where a 500 MB data is written to the disk array. The proposed architecture and RAID-10 have the same output rate with a peak speed of 3.75 MB/s. This rate is almost independent of the disk array size. In RAID-5, the overhead of excessive parity updates result in a lower output rate.

In RAID systems, the reliability of the RAID architectures can be also evaluated in terms of the maximum disk failure coverage, which means the maximum number of disk failures that can be tolerated by the disk arrays. Among the four architectures, RAID-5 can only tolerate the single disk failures and so it provides the lowest reliability. RAID-1 and RAID-10, due to the duplication of all primary disks, can tolerate up to $n/2$ disk failures. RAID-RMS, because of using a diagonal data distribution algorithm, provides higher reliability against both disk and data block failures. Therefore, we have conducted some simulation experiments to compare the reliability of RAID-RMS against the other mirroring-based architectures, namely RAID-1, RAID-10 in terms of data recovery rate. But, since RAID-1 and RAID-10 have the same results, Table 5 only shows a comparison between the proposed architecture and RAID-10. These results express the probability with which the blocks of a failed disk can be retrieved. In these experiments, the failures are generated with a uniform distribution within the disk arrays and the data recovery rate is measured as the

Table 3 – Scalability of RAID-RMS, RAID-10 and RAID-5 for read and write operations under a heavy I/O traffic condition (MB/S)

RAID architecture	Number of clients	Small read	Small write	Large read	Large write
RAID-10	2	1.83	2.05	2.00	2.12
	8	5.70	4.96	5.91	5.50
	16	9.14	9.17	10.12	9.49
RAID-5	2	2.11	1.21	2.15	1.85
	8	9.23	2.78	10.17	3.76
	16	16.02	4.86	16.11	5.86
RAID-RMS	2	2.40	2.12	2.27	2.23
	8	9.93	9.03	10.09	9.12
	16	16.24	15.38	16.15	16.01

Table 4 – The I/O rates of the proposed RAID architecture, RAID-10 and RAID-5 for read and write operations (MB/S)

RAID architecture	Number of clients	Small read	Small write	Large read	Large write
RAID-10	2	2.493	2.425	3.632	3.560
	8	2.752	2.516	3.782	3.728
	16	2.967	2.585	3.850	3.732
RAID-5	2	2.402	1.278	3.525	1.980
	8	3.122	1.509	3.787	2.085
	16	3.450	1.730	4.089	2.007
RAID-RMS	2	2.515	2.470	3.651	3.557
	8	3.140	2.835	3.825	3.750
	16	3.615	3.142	4.020	3.722

Table 5 – The disk failures recovery rate of RAID-RMS against RAID-10

RAID architecture	Number of disks	Number of failures							
		1	2	3	4	5	6	7	8
RAID-10	4	100	67						
	6	100	80	55					
	8	100	86	69	49				
	10	100	89	76	62	45			
	12	100	91	81	70	57	43		
	14	100	92	84	75	65	54	41	
	16	100	93	86	78	70	61	51	40
RAID-RMS	4	100	83						
	6	100	90	78					
	8	100	93	85	75				
	10	100	94	88	81	73			
	12	100	95	90	85	79	72		
	14	100	96	92	87	82	77	71	
	16	100	97	93	89	85	81	76	70

number of disks increases from 3 to 16 and the number of failures increases from 1 to $n/2$ in each experiment.

6. Conclusion

In this paper, we presented a new RAID architecture in which a special hybrid data distribution technique is used to locate the data blocks in the cluster. The main idea behind the proposed algorithm is to combine the data block striping and disk mirroring techniques with a data block rotation to improve the parallelism, reliability and efficiency of the RAID array. In such a RAID system, if a single disk fails, an effectively balanced read load can be attained at all surviving disks. It is shown that the proposed architecture is able to serve many more disk requests compared to the other existing mirroring-based RAID architectures. Besides, such architecture can tolerate more disk failures and decrease the rate of missing data due to the presented data distribution. So, this architecture can increase the parallelism for processing read requests, the reliability and the effective bandwidth to access data, especially after disk failures. The proposed architecture in terms of the maximum bandwidth, disk operation cost and reliability is theoretically analyzed and the obtained results show that this architecture outperforms the studied architectures in most of the investigated metrics and the experimental results completely confirm the analytic results. Finally, the given RAID architecture is appropriate to implement in the distributed storage systems which hold huge amounts of critical data.

Acknowledgement

The authors would like to thank the reviewers for their valuable comments and constructive criticism of the original manuscript.

REFERENCES

Akbari Torkestani J, Haghighat AT. ''A new redundancy algorithm for distributed environments.'' Operating system & Security Conference (OSSC), 2003, pp. 70–81.

Akbari Torkestani J, Meybodi MR. ''A new redundancy structure for using in high reliable systems.'' Proceedings of the Second International Conference on Information and Knowledge Technology (IKT2005), 2005a.

Akbari Torkestani J, Meybodi MR. ''A new data mirroring algorithm to enhance reliability and fault tolerance.'' Proceedings of the second international conference on information and knowledge technology (IKT2005), 2005b.

Akbari Torkestani J, Izadi T, Meybodi MR., ''Implementation of a new data security method for RAID''. Proceedings of 11th annual CSI computer conference of Iran, Fundamental Science Research Center (IPM), Computer Science Research Lab., 2006, pp. 157–64.

Anderson T, Dahlin M, Patterson D, Wang R. Serverless network filesystems. ACM Transactions on Computer Systems 1996: 41–79.

Asami S, Talagala N, Patterson DA. ''Designing a self-maintaining storage system.'' Proceedings of 16th IEEE Symposium on Mass Storage Systems, 1999, pp. 222–33.

Cabrera LF, Long DE. ''Using distributed disk striping to provide high I/O data rates.'' Proceedings of USENIX Computing Systems, 1991, pp. 405–33.

Cao P, Lim SB, Venkataraman S, Wilkes J. The TickerTAIP Parallel RAID Architecture. ACM Transactions on Computer System 1994;12(3):236–69.

Chen SZ, Towsley D. A performance evaluation of RAID architectures. IEEE Transaction on Computers 1996;45(10): 1116–30.

Chen PM, Lee EK, Gibson GA, Katz RH, Patterson DA. RAID: high-performance, reliable secondary storage. ACM Computing Surveys 1994;26(2):145–85.

Corbett P, Feitelson DG, Prost JP, Baylor SJ. ''Parallel access to files in the Vesta file system.'' Proceedings of Supercomputing'93, 1993.

Cormen TH, Kotz D. ''Integrating theory and practice in parallel file systems.'' Proceedings of DAGS '93 Symposium, 1993, pp. 64–74.

Cortes T. Software RAID and parallel filesystems. In: Buyya Rajkumar, editor. High performance cluster computing – architectures and systems. Prentice Hall PTR; 1999. p. 463–96.

Foster A, Kohr D, Krishnaiyer JR, Mogill J. ''Remote I/O: Fast Access to Distant Storage,'' Proceedings of the Fifth Workshop on I/O in Parallel and Distributed Systems, 1997, pp. 14–25.

Gibson G, Nagle D, Amiri K, Chang F, Gobioff H, Riedel E. ''A cost-effective, high-bandwidth storage architecture''. Proceedings of the 8th Conference on Architectural Support for Programming Languages and Operating Systems, 1998, pp. 97–106.

Harry M, Rosario JMD, Choudhary A. ''VIP-FS: a virtual, parallel file system for high performance parallel and distributed computing.'' Proceedings of the 9th International Parallel Processing Symposium (IPPS'95), 1995, pp. 159–64.

Ho R, Hwang K, Jin H. ''Design and analysis of clusters with single I/O space,.'' Proceedings of 20 International Conference on Distributed Computing Systems (ICDCS 2000), 2000, Taiwan, pp. 120–7.

Howard JH, Kazar ML, Menees SG, Nichols DA, Satyanarayanan M, Sidebotham RN, West MJ. Scale and Performance in a Distributed File System. ACM Transactions on Computer System 1988;6(1):51–81.

Hsiao HI, DeWitt D, ''Chained declustering: a new availability strategy for multiprocessor database machines.'' Proceedings of 6th International Data Engineering Conference, 1990, pp. 456–65.

Hsiao HI, DeWitt DJ. A performance study of three high availability data replication strategies. Journal of Distributed Parallel Databases 1993;1(1):53–80.

Hu Y, Yang Q, Nightingale T. ''RAPID-Cache a reliable and inexpensive write cache for Disk I/O Systems.'' Proceedings of the 5th International Symposium on High Performance Computer Architecture (HPCA-5), 1999, pp. 204–13.

Patterson DA, Gibson GA, Katz RH. ''A case for redundant arrays of inexpensive disks (RAID).'' ACM SIGMOD, Int. Conf. on Management of Data, June 1988, pp. 109–16.

Stonebraker M, Gerhard A, Schloss A. Distributed Raid – a new multiple copy algorithm. Berkeley, CA: University of California; 2000. 94720.

Tera data Corporation. ''DBC/1012: database computer system manual,'' Release 2, 1985.

Thomasian A, Blaum M. Mirrored disk organization reliability analysis. IEEE Transactions on Computers 2006;55(12):1640–4.

Thomasian. Shortcut method for reliability comparisons in RAID. Journal of Systems and Software 2006;79:1599–605.