A Layout Scheme for Duplicated RAID Systems

S.Subha
School of Information Technology
and Engineering
VIT, Vellore, India

ABSTRACT
Failure of disks in RAID is a bottleneck in processing. Data replication of RAID array systems is proposed in this paper for data validity. For a k-times replicated data in a RAID system with n disk arrays, the scheme takes the mean time between failures of the disks in a RAID system, allocates the data of the k-replicas in the n-RAID arrays based on the remaining time to the next failure and the distance of the disk array from the original copy of the data. The heuristics adapted places data in remote disks during the initial time period after a recovery and migrates the data to nearer disks as time advances to the next failure. A mathematical model is developed for the proposed scheme. Simulations support the proposed model.

General Terms
Algorithms, I/O

Keywords
RAID, performance, duplicate data

1. INTRODUCTION
RAID systems are vastly used for storage. Various types of RAID systems starting from RAID 0 to RAID n exist. The most common among them are RAID 0 to RAID 6. Further refinement in data storage can be done to obtain higher orders of RAID systems. The RAID system is bound to have a lifetime. During the lifetime it is liable to be under repair. A metric that measures the reliability of the RAID system is the mean time between failures (MTBF). The system initially is working well. As time approaches the MTBF, the system is prone to go down. It is repaired within the mean time to recover (MTTR) and is made functional again. Since RAID systems are prone to fail, backup of the data on a RAID system is needed. Distributed processing is the way of processing today. Almost all applications need resources from various locations and hence this finds application. In order to maintain the system up and running when a RAID system fails, it is replicated. Replications of the order of two or three are common. Research is going on in determining the way of replication needed for RAID systems so that data reliability, data availability is seamless. Authors in [4] have proposed data layout for RAID systems so that the whole system functions in spite of the primary data being out of reach due to the failure of the RAID disk. When the primary RAID system fails, the data is fetched from the remote RAID system. This involves some communication. Group rotate de-clustering [1], chained de- clustering [3] propose methods to store the replicated data in a particular fashion. Interleaved de- clustering was suggested in [2]. Certain de-clustering layouts were suggested in [5]. This paper proposes a method to place the replicated data of a RAID system so that data is available with minimum communication cost on a failure of the primary RAID system. By primary RAID system what the author means is the data on the original disk. An algorithm is developed to distribute the data of n-replications of a RAID system so that minimum time is spent in accessing data during the failure of the primary RAID system. The algorithm takes the MTBF of the RAID system as parameter along with the distances of the various remote places where the replications have to be present. A formula to place the data in a place that is inversely proportional to the time to next failure determined from the MTBF is developed. The data are placed suitably. The proposed model is simulated and the results support the model.

The rest of the paper is organized as follows. Section 2 gives the motivation, section 3 gives the algorithm, section 4 gives the simulation, section 5 gives the conclusion, and section 6 gives the references.

2. MOTIVATION
Consider a RAID system. Let the MTBF be one year i.e. twelve months. Let there be one replication of this system. Let the mean time to recover be one month. This is a pessimistic quantity. Assume the replications are placed in three different locations. During the one month to recover, the data has to be fetched from systems so that the total time for data access is minimized. Let there are 3 systems in the network located at distances in thousands of miles be 2, 10, 12. Let these systems be called A, B, C. Arrange the distances in ascending order of distances. The order is A, B, C. Consider the following allocation strategy. Divide the MTBF into three parts pertaining to the three locations. Find the fraction of the total time of MTBF with respect to the distance. Allot the duplication of the RAID system to the locations suitably. In the above example, the weighted ratio of distances with respect to the MTBF is given by (2/24)*12, (10/24)*12, (12/24)*12. These are equal to 2, 4, and 6. The units of measurement here are that of MTBF which is months. Thus after recovery or start of the system, for the first six months, the replication is placed in location C. It is placed in location B for the next four months and in location A during the last two months. When the system fails after twelve months, the data is fetched from location A which is the nearest system. As opposed to having the data in various locations and fetching it from them during a failure which involves much communication time, this algorithm saves time. This is the motivation of the paper.

3. PROPOSED ALGORITHM

Consider a RAID system i location in loc_i . Data is fetched from this system for computation. Let the MTBF for this system is m. Let the MTTR be r. Let the system be accessible in a network. Let there be n systems in the network. Let the distance i from these systems be given by $d_{11}, d_{12}, ..., d_{1n}$ where d_{ij} is the distance of the j-th system to system i. Hence the distance d_{ii} is zero. Let there be one replication of the RAID system. Arrange the distances of the n RAID systems from system i in ascending order. Let it be $a_1, a_2, ..., a_n$. For the system j, the time unit when the replicated system will reside in it after the i-the system recovers or starts is defined by the formula

$$\left(\frac{a_j}{\sum_{i=1}^{n} a_i} \right) m$$

(1) Thus the data in in the farthest location as soon the i-th system starts anew of after a recovery from a crash. This assumes that the MTBF is a reliable measure of the functioning of the system. The total communication cost during failure is gives as follows. If the volume of data to be accessed is v, and the distance of the nearest system is w, the total time taken is

vw+waiting time in queue at location w (2)

This is an improvement over fetching the data from remote sites farther than w. The waiting time in the system at location w is the bottleneck. Situations may arise when the waiting time is more so that the total time to transfer from some other location other than w is optimal. We categorize such cases as very rare and hence the above model.

The time complexity of the above algorithm is calculated as follows. It takes O(nlogn) for sorting the distances. It takes O(n) to calculate the time period of retention of the data in various locations. Hence the total time taken is O(nlogn+n). Now, assume that a new system is introduced into the network while the network is operational. The strategy followed is as follows. Let the new system be at location k. Find the distance d_{ik} . Let the replicated data be in location y and the next location where it has to be placed be x. Let the time for which the data is to be in location x be c_x. The following are the steps taken in this regard.

1. If $d_{ik} \succ d_{iy}$ then the new system is not included in the replication for the current operational time of the RAID system.
2. If $d_{ix} \prec d_{ik} \prec d_{iy}$ divide c_x among the locations x and k using their distances as criteria. Thus the two locations x and k will retain the replication for $\left(\frac{d_{ix}}{d_{ix} + d_{ik}} \right) c_x$ and $\left(\frac{d_{ik}}{d_{ix} + d_{ik}} \right) c_x$ respectively. Data access patterns refine this allocation. This is a topic for future research.
3. If $d_{ik} \prec d_{ix}$ the above steps can be repeated when condition given in (2) for any set of locations is matched. A special case is when the new system is nearer than the nearest network system. In this case, the time allotted to the nearest system is divided among the two systems using the same formula given in (2).

The case where many systems are introduced into the network can be extended with the logic given above.

Now consider the case where there are q replications instead of one. The network consists of n systems. The strategy is to place these q replications in these locations so as to have a non halting functioning of the network in case of the RAID failure in location-i. The strategy followed is as follows. The following cases occur.

1. q<n. Allocate the first replication according to the algorithm given above. For the rest of the q-1 copies, place them equidistant from location i.
2. q=n. The above algorithm can be implemented.
3. q>n. Allocate one additional copy at location i. The rest can be assigned arbitrarily in certain selected locations mostly that are nearer to location i but atleast one of them in a farther site that is not very much accessed. This facilitates smoother recovery.

4. SIMULATION

Consider a system at location i. Let the network consist of five systems A, B, C, D, E. Let the system in location i be A. Let the distances from A to other locations be 10, 15, 20, 30. Let there be a RAID system in location-i. Let there be three replications of it. Let the MTTF for the system in location i be 25 months. The amount of time that the replication #1 will stay in various locations is given as follows.

Location	Distance	Time in Months
A	0	0
B	10	3.333333
C	15	5
D	20	6.666667
E	30	10
Sum		25

There are two more replications. For replication #2 allot it to location A. The third replication assigns it to location E as it is not used mostly. The performance improvement that is gained by this strategy is as follows. Consider the system where the replication is not at the nearest system at the point of failure. Then communication overhead increases. This is reduced in this case. When all the nearest replication fails, the replication for location E can be used. The algorithm mentioned in [6] can be extended to allot the replicated RAID among the various locations to save energy consumed.

5. CONCLUSION

RAID systems are vastly used in present day applications. The reliability of the RAID system has a major say in the functioning of the system. RAID systems are duplicated to recover from failures. An algorithm to recover from RAID system failure is presented in this paper. An example is given to simulate the algorithm and it is found to perform better as a function of the communication cost of the network.

6. REFERENCES

[1] S.Chen, D.Townsley. A Performance evaluation of RAID architectures, IEEE Trans. Comput., 45(10): 1116-1130, 1996

[2] G.Copeland, T.Keller. A comparison of high-availability media recovery techniques, SIGMOD'89

[3] H.L.Hsiao,D.J.DeWitt. Chained declustering: A new availability strategy for multiprocessor database machines. In Proceedings of the Sixth International Conference on Data Engineering, pages 456-465

[4] Hujin Zhu, Peng Gu, Jun Wang. Shifted De-clustering: A Placement-ideal Layout Scheme for Multi-way Replication Storage Architecture, Proceedings of ICS'08

[5] D.A.Patterson, G.Gibson, R.H. Katz. A case for Redundant Arrays of Inexpensive Disks (RAID), In Proceedings of 1988 ACM Conference on Management of Data (SIGMOD), pp-109-116

[6] S.Subha. A Disk Allocation Algorithm, PDPTA,'07