データベースシステム 104－7
（1995. 7. 18）

# Hot Mirroring を用いたディスクアレイの基本性能評価

茂木 和彦 喜連川 優

東京大学 生産技術研究所
〒106 東京都港区六本木 7-22-1

あらまし

近年、高性能・高信頼性の2次記憶装置としてRAIDが注目を集めている。小さなアクセスサイズのものが多数要求される負荷に対しては、ミラー(RAID1)とRAID5が適していると考えられる。RAID5はパリティによる冗長化を行っているため、データ更新時のオーバヘッドとディスク故障時の性能の低下という問題が存在する。一方、ミラーは全てのデータのコピーを保持するため、ミラーにはRAID5と比較してデータ容量がかなり小さく抑えられるという欠点が存在する。通常、アクセスには偏りが存在すると考えられる。そこで、アクセス頻度が高いのもはミラー化、アクセス頻度が低いものはパリティによる冗長化を行うことにより、双方の長所を採り入れようと試みる "Hot mirroring" と名付けた手法を提案する。本論文では Hot mirroring 方式の有効性を検討する。

キーワード： ディスクアレイ、RAID5、ミラー、2次記憶装置

# Preliminary Evaluation of Hot Mirroring

Kazuhiko MOGI Masaru KITSUREGAWA

Institute of Industrial Science, University of Tokyo
7-22-1, Roppongi, Minato-ku, Tokyo 106, JAPAN.

Abstract

Recently RAID has attracted strong attention as a high performance and high reliable secondary storage system. For the loads which consist of a large number of small accesses, RAID level 1 (mirrored disk arrays) and RAID level 5 (RAID5 disk arrays) are the best suited among several RAID levels. The major drawback of RAID5 disk arrays is, however, in the large overhead incurred for small writes and the significant performance degradation on disk failure, which is caused by introducing parity encoding for redundancy. Mirrored disk arrays have also problems. They have considerably smaller storage capacity than that of RAID5 disk arrays because they make a copy of all the disk blocks for redundancy. In order to get not only higher performance but also larger capacity, we propose yet another storage scheme named "hot mirroring". There is alway localities in the disk accesses. We store those hot data on mirrored disk. Non-frequently accessed data is stored in RAID5. In this paper we examine the feasibility of hot mirroring.

Key Words : Disk Array, RAID5, Mirror, Secondary Storage

# 1 Introduction

Recently RAID[1] has attracted strong attention as a high performance and reliable secondary storage system. RAID utilizes a large number of commodity inexpensive drives in parallel to achieve higher performance as well as obtaining higher reliability by recording redundant informations. In [1], Patterson et. al classified RAID into five levels. Among five levels, level 1 (mirrored disk array) and level 5 (RAID5 disk array) are regarded as one of the most promising approaches for providing highly reliable secondary storage systems which support concurrent access of small blocks, such as file servers. Mirrored disk arrays make a copy of all disk blocks for redundancy. In contrast, RAID5 disk arrays employ parity encoding for redundancy, which leads to much larger storage capacity than that of mirrored disk arrays when the same number of disks is used.

There are two big problems in using RAID5 disk arrays. One is the overhead of recording redundancy information. The new parity for a small write is derived as follows:

$$P_{new} = P_{old} \oplus D_{old} \oplus D_{new} \quad (1)$$

Thus a single block update requires 4 disk accesses: old block read ($D_{old}$), old parity read ($P_{old}$), new block write ($D_{new}$) and new parity write ($P_{new}$). This deteriorates the throughput of the write operations. The other problem is the overhead of reconstructing data when some disks fail. When disk $k$ in parity group $j$ fails, the lost data is rebuilt as follows ($D_{j,i}$ means the data location on disk $i$ in disk group $j$ in which the parity stripe was made):

$$D_{j,k} = P_j \oplus D_{j,1} \oplus \cdots \oplus D_{j,k-1} \oplus D_{j,k+1} \oplus \cdots \oplus D_{j,n} \quad (2)$$

Thus the rebuild process requires disk accesses for the all disks in the disk group of the failed disk. The impact of this operation on performance is quite large.

In mirrored disk arrays, a copy is stored for redundancy. In normal mode, only two write accesses are required for a block update. During rebuild mode, only a read access on the live disk and a write to the new disk are required for the replacement of a failed disk. Therefore the overhead on a block update and the load of rebuild process are much smaller than that of RAID5 disk arrays. Moreover, using the copy, mirrored disk arrays can easily balance the load amongst their disks not only in normal mode but also during rebuild mode. But described before, mirrored disk arrays pay the penalty of much smaller data capacity than that of RAID5 disk arrays, because of the data copying.

Usually there are localities in the access pattern. Exploiting the access localities, the hot block clustering method[2] separates data into two groups, one is the group of data which have high access rates and the other with low access rates. We showed this separation leads to an improvement in performance for dynamic striping RAID5 disk arrays[3]. In order to get not only higher performance but also larger usable capacity, we consider the combination of a mirrored disk array and a RAID5 disk array with hot block separations which are used in the hot block clustering method. We divide each disk into two contiguous regions, a hot region and a cold region. The data on the hot group is mirrored to decrease the overhead of maintaining redundancy information and the rebuilding data and to balance the load amongst each disk. For the cold group, we use parity protection for redundancy to obtain higher storage efficiency. We name this storage management scheme "hot mirroring".

To examine the feasibility of hot mirroring, the performance for normal mode and the rebuild mode are analyzed through simulation. For high access locality, higher performance as compared with RAID5 disk arrays is obtained. For low access locality, the gain of this method decreases but higher performance can still be archived.

# 2 Hot Mirroring

## 2.1 Concept of hot mirroring

With respect of high storage efficiency with high reliability, RAID5 is best among all RAID levels. But there are two big problems with using RAID5 disk arrays for high performance storage systems. One is the overhead time required to record redundancy information. The other is the overhead time of reconstructing data when some disks fail. It is the parity encoding for redundancy that causes these problems. From point of view of performance, mirrored disk arrays, which

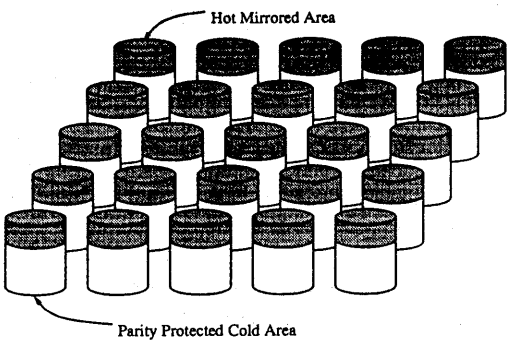



Figure 1: Hot mirroring

use block copying for redundancy, is better than RAID5 disk arrays. But mirrored disk arrays have significantly large space overhead. If it were possible to merge the characteristics of high storage efficiency from RAID5 disk arrays with the low overhead of recording redundancy information from mirrored disk arrays, it might be one of the best configurations of disk arrays.

In general, there are access localities, which can be utilized to improve the performance of RAID5 disk arrays. To solve problems of mirrored disk arrays and RAID5 disk arrays, access localities are exploited. According to the access frequency, two groups of disjoined blocks are made, one group contains blocks with high access rates (hot blocks) and the other has low access rate blocks (cold blocks). With this separation, the mirror scheme and the parity encoding scheme are combined to get higher performance and larger capacity.

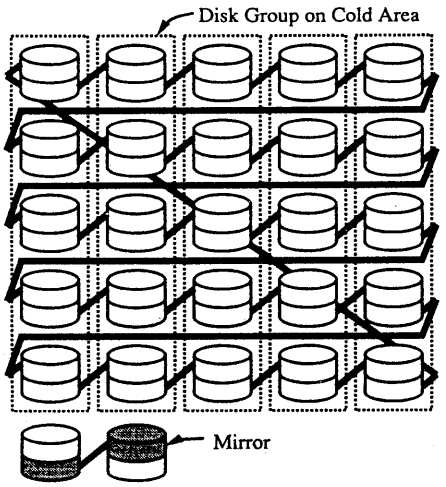
For load balancing, it is important to distribute the access requests for hot blocks evenly over all disks. It is also desirable that the penalty of recording redundant information on hot blocks be small. As mentioned before, these requirements are well suited to the mirror scheme, thus hot blocks are mirrored. For the cold area, we use parity protection for redundancy to obtain higher storage efficiency. Moreover the penalty for maintaining redundant information in the cold area has little effect on performance because these blocks are infrequently updated. This storage management scheme is called "hot mirroring" (figure 1).

Identification of hot block is key to this proposed method. Usually almost all blocks written to by write requests can be regarded as hot, since these blocks tends to be used again. So we assume that all blocks of normal write requests are hot and execute all write accesses to the hot area. With small probability, write operation is done against cold block. Thus our strategy writes cold blocks to the hot area on write operation, which consumes the free space on the hot area. Cold blocks in the hot area need to be migrated back to the cold area. By recording the time at which each block in the hot area was last accessed, the cold blocks residing in the hot area can be found by finding the blocks with the oldest access time. If the amount of free space in the hot area falls below the threshold value, this migration is invoked. Cold block migration from hot to cold needs two extra write accesses and a read access for cold block write operations. As will be clarified in the section 4, this is not a fatal overhead.

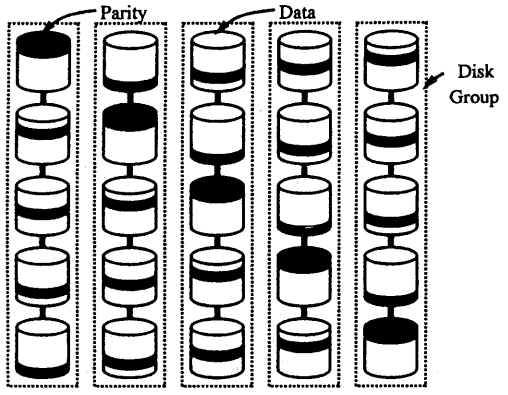
## 2.2 Data placement policy

The data placement policy of the mirrored hot area and parity protected cold area considerably impacts performance during rebuild mode because the effect of the rebuild process on the parity protected area is very large. For hot mirroring, the copy allocation of the hot area is illustrated in figure 2(a) and parity stripes for the cold area are as shown in figure 2(b). In the cold area, parity stripes are made into a disk group (vertically in the figure). In the hot area, the copy is allocated on different disk group (horizontally in the figure).

The reason why we employ such orthogonal placement for parity stripe and mirroring is as follows. Since the rebuild time needs to be minimized, the disks of the parity stripe including broken disk should work for rebuilding as solely as possible. This means that frequent accesses against hot areas of broken disk group should be toward to the mirrored hot area on the other disk group. Thus parity stripe and copy allocation is orthogonal each other. By employing this scheme, hot accesses can be served by surviving stripes, while all the drives on the broken stripe works for rebuilding. In addition, the hot accesses against mirrored area can be absorbed without degradation by distributing the traffic equally among the remaining stripes employing




(a) Hot area




(b) Cold area

Figure 2: Data allocation policy

the following mirroring scheme. Hot mirroring distributes the copy on the hot mirrored area as shown in figure 2(a), which is based on the chained declustering method[4].

# 3 Simulation Experiments

## 3.1 Implementation issues

**Floating block allocation in the hot mirrored area and read/write operations** In this simulation, hot mirrored area acts as a large cache for the cold area. That is, when the frequent accessed blocks are migrated from parity protected area onto the mirrored area, the space initially allocated on the parity protected area is not discarded and its location is fixed. This means that hot blocks occupy three times as large space as the original size.

The position in mirrored area is not fixed. Floating scheme[5] is employed. On write operation, the algorithm of determining the location for the new hot block over chained declustered mirrored array is as follows. Here we assume that each disk has the associated access request queue. To balance the load, the length of access queue on each disk is checked.

1. Find the shortest length access queue amongst all disk pairs containing free blocks. Here, the length of the access queue in a disk pair is defined to be the longer queue length in that pair.
2. If the requested block is already in the hot area and the shortest length is longer than the length of the original disk pair, data is written to the previous position.
3. Otherwise find the disk pair whose number of free blocks in the hot area is the largest.
4. Choose the new position from the free blocks in the selected disk pair. (In this simulation we selected new position randomly for the simplicity.)

For reading, the disk which has the shortest access queue is selected.

If the read/write is issued against cold area, these requests are served on its original position since the hot mirrored area does not have a copy.

**Rebuilding process** The method of dispatching access requests for data rebuilding has a lot of impact on the performance on the rebuild mode. As described in [6], the unit of reconstructing data affects the performance of the rebuild mode and the rebuild period. In this simulation, we adopt the track as the unit of rebuilding in consideration of efficiency in the rebuilding process and the impact to normal requests. We regard a track access for rebuilding as two access requests when we count the length of access queue for load balancing. Reads for the reconstructed data are redirected to balance the loads. For the parity protected area, live data from the failed disk group is read when both read and write accesses to blocks on the failed disk are requested. Thus reconstruc-

tion of the track containing a requested block is always performed in the parity encoded region. On the other hand, there is no need to reconstruct data in the mirrored area when a broken block on the mirrored area is requested. In this simulation, if there are no more than two read access requests to the ones being rebuilt on the pair disk, piggybacking is performed because the overhead of piggybacking on a track is small for the read operation. However write requests to the failed disk are not performed in the standby disk because the overhead of reading old data on write accesses is larger than that for read accesses.

A baseline rebuild is performed. For the parity protected area, we make a new rebuild access request if there is at least one live free disk in a broken disk group and the number of rebuild read requests on all disk's access queue is no more than two requests. For the mirrored area, we make a new rebuild access request when broken disk's paired disk is free and the number of rebuild read accesses in the access queue is no more than two. But in both areas, a new rebuild request is not performed if there are more than ten access requests on the repair disk, except for the following exception.

For a speedy rebuild, the rebuild time is limited. A constraint placed on the rebuild process is that the number of rebuild blocks must exceeded the elapsed time multiplied by a constant. If insufficient blocks have been rebuilt, dispatch baseline rebuild access requests are forced. In this simulation, the maximum rebuild time is 40 minutes (about 10 times of disk full scan time).

### 3.2 Simulation assumptions

Simulation parameters are as follows. Table 1 shows the disk model parameters. The block size is 4KB. The striping unit is set to the block size. The position of the parity is incremented by one track when rotated among the disks of the parity protected cold area.

To compare performance, four configuration are examined, hot mirroring, naive RAID5, mirroring with fixed data position, and mirroring which adopts data floating and uses the same method to balance the load as hot mirroring on write operations. Naive RAID5 is the same management scheme as used by the cold area management on hot mirroring, and mirroring and mirroring with data floating disk arrays uses the same management scheme of hot mirrored area on hot mirroring. Table 2 shows the effective data capacity of these configurations in the simulation.

| | |
|---|---|
| cylinders/disk | 949 |
| tracks/cylinder | 14 |
| sectors/track | 6 |
| sector size | 4096 bytes |
| revolution time | 13.9ms |
| seek time model | $\text{seek}(d) = 2.0 + 0.01 \cdot d + 0.46 \cdot \sqrt{d}$ |
| track skew | 1 sector |

Table 1: Disk model parameters

| | |
|---|---|
| RAID5 | 83.3 % |
| Mirror (naive) | 50.0 % |
| Mirror (data floating) | 49.9 % |
| Hot Mirroring | 66.7 % |

Table 2: Data capacity on each configuration (normalized by total disk volume)

Disk arrays are composed as follows: Hot mirroring has 4 disk groups, which have 5 data disks and a parity disk in the cold area. (4*(5D+P) denotes this parity configuration later.) 20% of the total physical disk capacity are allocated to the hot mirrored area, which means that 15%[1] of the total blocks in the cold area can be stored in this area. The hot area must have some amount of free area on each disk pair. This free capacity is set to half a cylinder. In other word, when the number of free blocks in the disk pair becomes less than half a cylinder, data migrations are executed. The naive RAID5 disk array has 4*(5D+P) configuration. In this simulation, data position is fixed and 20% of total capacity is not used at all for the adjustment of the capacity for hot mirroring. Both naive mirrored disk array and data floated mirrored disk array adopt the chained declustering method. Mirrored disk arrays have the same number of disks as used for hot mirroring in order to compare performance. Therefore, the data capacity of mirrored disk arrays is smaller than that of hot mirroring. The data floated mirrored disk array must have some free area. This free capacity is set to half a cylinder for each disk pair.

To simplify the simulation, it is assumed that the disk array controller and the bus between the

[1] $\frac{0.2/2}{0.8 \times 5/6} = 0.15$

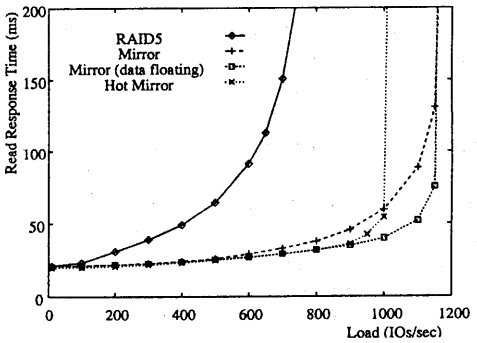
controller and disks are sufficiently fast. Based on this assumption, the controller can find free disks and dispatch accesses for rebuilding as soon as some disks become free. All the control tables are maintained by the controller. Disks also have an intelligent controller and begin track accesses on the sector which the disk head encounters first after the head becomes available.

Disk accesses (which includes migration accesses and rebuild accesses) are performed on first come first serve basis. Access requests are fixed at 4KB. The interval of access request arrivals have a negative exponential distribution. The load is controlled by changing the mean time between access requests. That is, the access requests are randomly distributed. Access locality is as follows: blocks are divided into two groups and $y$% of the blocks belong to the first group. In each group, the access probabilities are equal, but $x$% of the access requests are concentrated on the first group. (Later referred to as $x$-$y$ access locality.) The blocks which belong to the first group are randomly distributed over all the disks. Statistics gathering begins after initial 2 millions write accesses to hot mirroring and after initial 100 thousands write accesses to the other disk arrays.

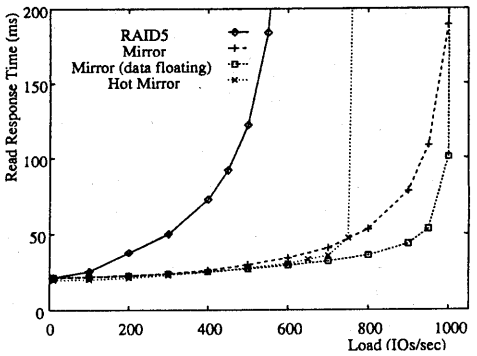
# 4 Evaluations of hot mirroring

## 4.1 Read response time analysis in normal mode

Figure 3 shows the response time in which 90% of the read requests have been completed for 100,000 access requests for 90-10 access locality for two different write ratios ( (a) Read:Write = 7:3, (b) Read:Write = 1:1 ). The horizontal axis shows the mean arrival rates of I/O requests, the vertical axis shows the read response time. Hot mirroring shows much better performance than the naive RAID5 disk array. At low loads, hot mirroring shows almost the same performance as the mirrored disk array with data floating and a little better performance than the mirrored disk array. But hot mirroring cannot bear higher loads than mirrored disk arrays because the separation cost becomes non-negligible for high loads. This overhead is required for the write operation. Therefore the higher the write probability becomes, the worse performance of the hot mirroring will be relative to mirrored disk arrays.




(a) Read : Write = 7 : 3




(b) Read : Write = 1 : 1

Figure 3: Response time for 90% of the read requests in normal mode (90-10 access locality)

## 4.2 Read response time analysis during rebuild mode

The performance during rebuild mode is as important as that of normal mode. Figure 4 shows the response time for completion of 90% of the read requests for access requests during the rebuild process with 90-10 access locality on Read:Write = 7:3. Every method shows slightly worse performance than that of normal mode. Among all configurations, the naive RAID5 disk array is most strongly affected by the rebuild process. The curve of rebuild mode is shifted upward compared with that of normal mode. In this simulation, the access probability for a disk during normal mode is about 4.2%. So 90% completion response time shown in figure 4 may show the value for blocks which are on live disks.

In order to clarify the impact on the access requests which are highly affected by the rebuild

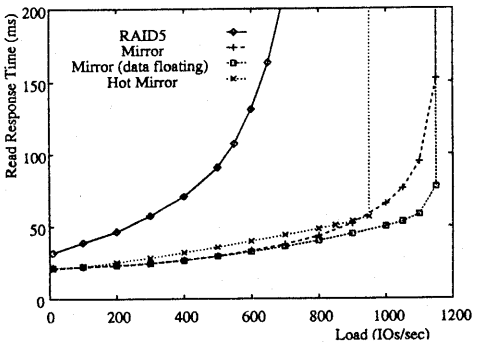



Figure 4: Response time for 90% of the read requests during rebuild mode (90-10 access locality, Read:Write = 7:3)

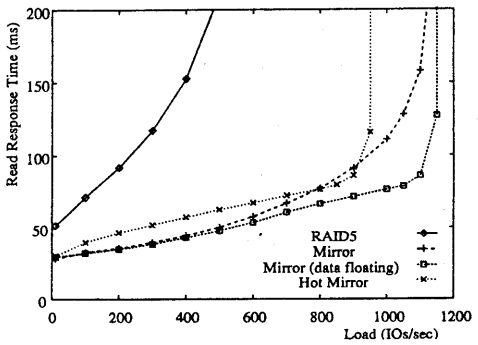



Figure 5: Response time for 99% of the read requests during rebuild mode (90-10 access locality, Read:Write = 7:3)

process such as requests to the broken disk, the response time for completion of 99% of the read requests on the same data used in figure 4 is examined. Figure 5 shows the result. The naive RAID5 disk array is strongly affected by the rebuild process. The other methods show slightly worse performance than that of 90% read requests case. Hot mirroring shows worse performance than that of mirrored disk arrays. In mirrored disk arrays, all data is copied. In hot mirroring, reconstruction of the broken data in the cold area is required. This difference causes the response degradation for hot mirroring. But the performance of hot mirroring is significantly better than that of the naive RAID5 disk array since most of the read requests against broken disk can be covered by paired drive in hot mirroring.

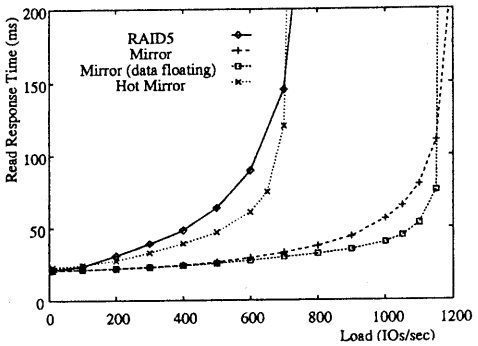


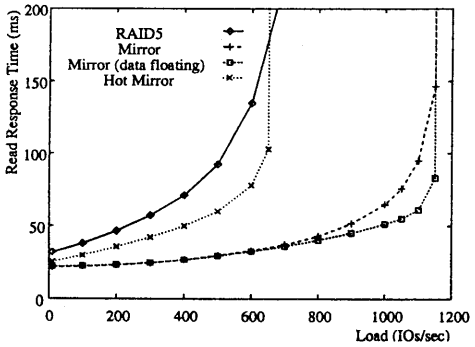

Figure 6: Response time for 90% of the read requests in normal mode (80-20 access locality, Read:Write = 7:3)

### 4.3 Impact of access locality

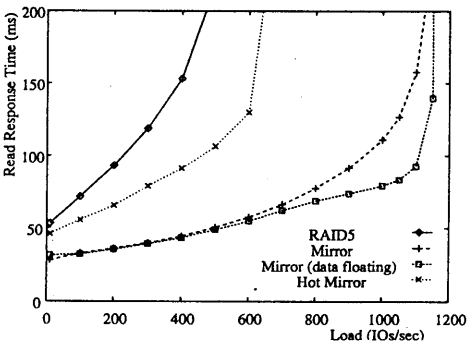
Hot mirroring makes use of access locality for improving performance. The degree of access locality may affect performance. Figure 6 shows the read response time for completion of 90% of the read requests complete for 100,000 access requests with 80-20 access locality on Read:Write = 7:3 in normal mode.

For 80-20 access locality, hot mirroring shows much worse performance than does 90-10 access locality and somewhat better performance than that of naive RAID5 disk arrays. In this simulation, the disk array which adopts hot mirroring holds the hot area whose capacity is 15% of the total data capacity. For 90-10 access locality, all hot blocks are stored in the hot area. But for 80-20 access locality, not all of the hot blocks can be stored in the hot area, which causes block migrations to occur more frequently. Therefore the overhead of separating hot blocks becomes much larger for 80-20 access locality than for 90-10 access locality. Thus the effectiveness of hot mirroring decreases when the access locality is not high, but it never shows worse performance than naive RAID5 disk arrays.

Figure 7 shows the performance for 80-20 access locality during rebuild mode. Although the overhead for write accesses is large, hot mirroring can still balance the load to make use of the copy in the hot area. So hot mirroring shows better performance than that of naive RAID5 disk arrays during rebuild mode also.




(a) Response time for 90% of the read requests




(b) Response time of 99% of the read requests

Figure 7: Read response time during rebuild mode (80-20 access locality, Read:Write = 7:3)

## 5 Conclusion

This paper presents the new storage management scheme named "hot mirroring" for obtaining higher performance and larger data capacity. This scheme makes use of access localities. Each disk is divided into two regions, the hot area and the cold area. In order to reduce the overhead of recording redundant information and to balance the load among all disks, all blocks in the hot area are mirrored. In the cold area, a parity encoding scheme is adopted for redundancy with low storage overhead. Hot mirroring makes the assumption that all written blocks are hot. Cold blocks in the hot area are estimated by examining the elapsed time since the last access occurs. They are migrated to the cold area according to the number of free blocks in the hot area.

The feasibility of hot mirroring was examined through simulation. For high access localities, hot mirrored disk arrays show much higher performance than that of naive RAID5 disk arrays. At low loads, hot mirrored disk arrays have slightly better performance than the mirrored disk arrays and almost the same performance as that of mirrored disk arrays which adopt data floating. But hot mirroring cannot provide higher performance than mirrored disk arrays because of the overhead of separating the hot blocks. During rebuild mode, hot mirroring shows slightly worse performance during the rebuild process than do mirrored disk arrays, but has much better performance than naive RAID5 disk arrays. For low access locality, the overhead of separating hot blocks and cold blocks becomes high but hot mirroring never becomes worse performance than that naive RAID5 disk arrays.

In this paper, we intended to clearly show the feasibility of using the hot mirroring method. Therefore we did not combine other methods which have been proposed to improve performance which could increase further hot mirroring's performance. There is also room for improving storage efficiency. These optimizations are for future investigation.

## References

[1] D. A. Patterson, G. Gibson, and R. H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proc. of ACM SIGMOD*, pp. 109–116, Jun. 1988.

[2] K. Mogi and M. Kitsuregawa. Hot Block Clustering for Disk Arrays with Dynamic Striping — exploitation of access locality and its performance analysis . *Will appear in 21st VLDB*, Sep. 1995.

[3] K. Mogi and M. Kitsuregawa. Dynamic Parity Stripe Reorganizations for RAID5 Disk Arrays. In *Proc. of 3rd PDIS*, pp. 17–26, Sep. 1994.

[4] H. Hsiao and D. DeWitt. Chained Decrustering: A New Availability Strategy for Multiprocessor Database Machines. In *Proc. of IEEE Data Engineering*, pp. 456–465, Feb. 1990.

[5] J. Menon and J. Kasson. Methods for Improved Update Performance of Disk Arrays. In *Proc. of HICSS*, volume I, pp. 74–83, Jan. 1992.

[6] R. Y. Hou, J. Menon, and Y. N. Patt. Balancing I/O Response Time and Disk Rebuild Time in a RAID5 Disk Array. In *Proc. of HICSS*, volume I, pp. 70–79, Jan. 1993.