Using Active Data Repair and Migration Technologies for RAID-based Storage Systems

Yin Yang
School of Accounting. Wuhan Textile University, Wuhan 430200, China
Email: cs_yangyin@hust.edu.cn

Wei Liang*
The 722 Institutes of China Shipbuilding Industry Corporation, Wuhan 430070, China
Email: liangwei19830725@gmail.com

Zhihu Tan and Jiguang Wan
School of Computer Sci. and Tech. Huazhong University of Sci. and Tech, Wuhan 430074, China
Email: {stan, jgwan}@hust.edu.cn

Sandeep Subhoyt
School of Computer Science. Virginia Commanwealth University, Virginia VA 23284, USA
Email: subhoyts@vcu.edu

***Abstract*—This paper proposes a new RAID-based system using Active data Repair and Migration technologies called RARM, which detects the status of sector, then allocates reserved space in every disk and arranges to redirect the data in the non-repairable sectors to the reserved space. If we ensure the reliability and utilization of disk, RARM will copy data of the low reliability disk or the number of the bad sectors in the disk exceeding threshold to a new one; it will also copy and redirect popularity data of the medium reliability and high utilization disk to the reserved space in high reliability disk or a new one, these process, as we called data migration can avoid lengthy data reconstruction. It can adjust migration speed to reduce the impact on the front end performance dynamically and prioritize user requests when the system I/O workload is bursting. The overall results indicate that the RARM can improve performance and reliability of the system with little influence.**

***Index Terms*—RARM, RAID system, reliability, data repair, data migration**

I. Introduction

With the rapid development of science and technology, internet technology and related network applications, the employed and processed object of information presents explosive growth. As the carrier of information, the quantity of data has been ceaselessly increasing. Data contains abundant information and value which have become one of the most valuable treasures of human beings. Data is a significant safeguard and impetus basis for the survival and development of human beings. The value of data in storage system has far surpassed the price and value of the storage system hardware. To enterprises, data loss would cause inevitable and incalculable damage, inconvenience for producers' and consumers' work and also irreparable and unimaginable material damage; what's worse, it may even destroy the life of the enterprise.

All in all, faced with rapidly increasing quantity of data and the ideology of irreplaceable data, it raises new challenges for storage systems and relevant technologies. How to improve reliability has already become the focal point to the enterprise users [23]. In order to address these problems, the redundancy or backup methods [5, 7, 33] have been presented. When the disk fails, the missing data can be recovered according to the redundant information, this technology called RAID [1]. When there is failure in the disks, the RAID will fall into the degradation mode, the system then choose a new disk to carry on the data reconstruction, but this process needs massive disk I/O operations, in the event of a second disk failed, it will overwhelm tolerance ability of the RAID, and will thus cause the data loss [2].

It is important to note that disk failures do not always happen instantaneously, but it is possibly the partial recoverable medium error [31]. And many applications have one notable feature [11]: a small portion of the entire dataset is accessed much more frequently. This portion forms a popularity of the dataset [12]. Therefore, how to efficiently migrate that popularity in medium reliability disk is critical to improve data reliability [32, 34].

To solve the data part medium error, predict disk failure and protect popularity data, this paper proposed a RAID system with active data repair and migration technologies called RARM, a smart system just like smart RAID system [3, 4]. RARM divides storage system into four parts: sector, low reliability disk, medium reliability disk and high reliability disk, and statistics the

* Corresponding Author

doi:10.4304/jcp.9.10.2281-2293

utilization of disk and predicts the popularity of data. It aims to resolve the partial sector error through bad sectors redirection, simultaneously restoring data of the faulty section, when the bad sector surpasses the threshold or the low reliability disk is confirmed. RARM will initiate the migration of the data in these disks to the idle disk, and it will also copy and redirect popularity data of the medium reliability and high utilization disk to the reserved space in high reliability disk or a new one, thereby avoiding the effect of performance caused by reconstruction operations. In this paper, we use disk SMART technology to ensure the reliability of disk and confirm the reliability of disk through comparing the output value of early-warning threshold. It also uses Zipf-like distribution technology and the popularity data rank at the present time to dynamically predict new popularity data. So we can get the utilization of disk at the present time and new popularity data rank in the future time. In this paper, we detailed discuss the function realization of data migration.

We also detailed discuss the new added issues in data migration functional module: data migration information management; RAID workload testing; synchronization unit and data consistency.

In numerical results and discussions, we evaluate the data migration velocity of traditional normal RAID and RARM with Iometer (RAID5 and RAID1). We also give the practical application results using the benchmarks when disk has failed in RAID systems. We measured both front and end performance in terms of average user response times while online recovery operation and the total online recovery times for the three benchmarks.

The rest of this paper is organized as follows. We first describe fundamental principle of RARM in Section II. Section III introduces the software structure about RARM and comprehensively discusses the structure of RARM system functional module in Section IV. Section V evaluates the RARM system and provides a detailed analysis of results and presents related work in Section VI. Finally, our conclusions are drawn in Section VII.

II. The General Outline Of RARM

In this section, we first introduce the data repair technology. Second, we reveal the data migration technology to reduce the data recovery time.

A. Data Repair Technology

In RARM, data repair technology will reserve some sector space in each disk. RARM first detects the bad sectors. Second, it automatically allocates the equivalent space from reserved space and replaces the bad sectors. Final, RARM writes data to allocated position.

The data block storage of a 4-disk RAID5 as shown in Fig. 1. Every disk is divided into many blocks, which is used to store actual data and redundancy information. For example, P0, P1, P2 and P3 are blocks of parity information. A, B and L stand for data blocks. Blocks of parity information are generated through RAID5 parity generation algorithm in the unit of stripe. In Fig. 1, every stripe consists of 4 blocks. For instance, stripe zero consists of P0, A, B and C, and P0=A+B+C [6]. Because it stores the parity information, if any block of data is lost, it can be regained by the data stored in the other three blocks.

	Disk0	Disk1	Disk2	Disk3
Stripe1	P0	A	B	C
Stripe2	D	P1	E	F
Stripe3	G	H	P2	I
Stripe4	J	K	L	P3
Stripe5	Blocks	Blocks	Blocks	Blocks
	⋮	⋮	⋮	⋮
Stripe	Blocks	Blocks	Blocks	Blocks

Figure 1. RAID data block storage.

So, if a bad sector appears in RAID system, it first recovers bad sectors, then puts other blocks in the same stripe into memory with the parity information, and finally recovers the data utilizing reconstruction of data algorithm according to the level in RAID that the disk belongs to and stores the data into the sector which replaces the bad sector.

In order to improve the data reliability further, RARM distributes many reserved space to other disks, or even replicate more copies as described in Fig. 2, disk a and disk b in the RAID all have 3 bad sectors, the recovery data of sector labelled 1 has two copies, one is written to its reserved space belonging to the source disk, the other to the adjacent reserved space, the rest have the same operation until all bad sectors are recovered [31]. This means it can improve the reliability of RARM system significantly, where the reserved space can give higher level warning, triggering the disk data at a higher level.

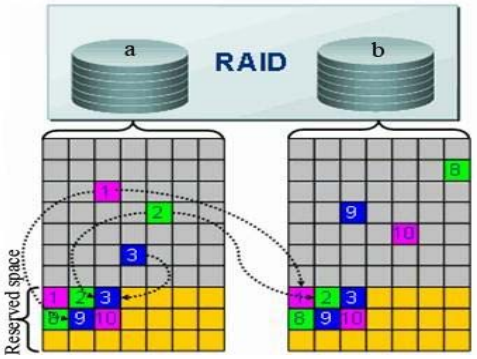


Figure 2. Data repair technology [31].

B. Data Migration Technology

When the number of bad sectors in the disk exceeds the threshold, or one disk is medium reliability and high utilization have popularity data, or one disk is low reliability, the data or popularity data in the specified disk should initiate migration to other idle and fit disk to maintain the reliability. It will copy data of the low reliability disk or the number of bad sectors exceeds the threshold in one disk directly to a new disk quickly at the right moment. If we ensure the reliability and utilization of disk, it will also copy popularity data of the medium reliability and high utilization disk to the reserved space in high reliability disk or a new one, as shown in Fig. 3.

Data migration will also redirect disk I/O requests to the reserved space if query operations get a positive answer.

In Fig. 3, we assume that the low reliability disk is D5; the medium reliability disks are D1, D2 and D3, in which D1 also is high utilization disk; and the high reliability disk is D4. At the present time, the popularity data includes: d1, d2, d6 and d12. In the future time, the new popularity data includes: d1, d5 and d11.

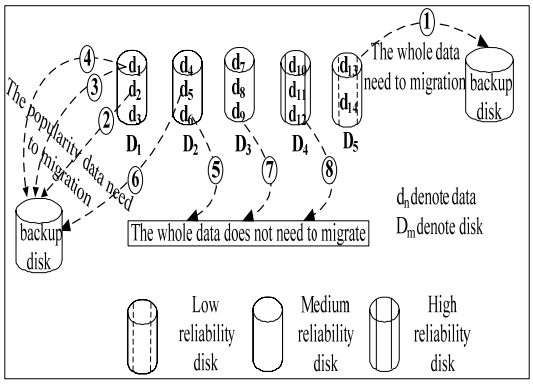


Figure 3. Data migration technology.

We categorize the reliability of disk in three levels: low reliability disk, medium reliability disk and high reliability disk [30]. If the disk is medium reliability disk, it may become the low reliability disk in two ways [13]: the medium reliability disk is in high utilization and still in use from the present time to the future time. We categorize the utilization of disk in terms of popularity data rank at the present time in three levels: low, medium and high. The detail of realization steps, as shown in Fig. 4.

Algorithm: The conditions to trigger the data migration mechanisms

Step 1
if: the disk is low reliability disk;
then: the whole data is migrated to backup disk;

Step 2 and Step 3
if: the medium reliability disk is also high utilization disk, we confirm the popularity data is in this disk in terms of popularity data rank at the present time, and this disk has popularity data at the present time;
then: migrate the popularity data to backup disk;

Step 4
if: the medium reliability disk is also high utilization disk, we prediction the new popularity data is in this disk in the future time in terms of popularity data prediction technology, and this disk has new popularity data in the future time;
then: migrate the popularity data to backup disk;

Step 5
if: the medium reliability disk is not high utilization disk;
then: the popularity data at the present time does not need to migrate;

Step 6
if: the new popularity data predicted in the future time;
then: the new popularity data migrate to backup disk;

Step 7
if: the medium reliability disk has no popularity data both at the present time and in the future time;
then: the whole data does not need to migrate;

Step 8
if: the disk is high reliability disk;
then: the whole data does not need to migrate;

Figure 4. The detail of realization steps for data migration technology.

We use disk SMART technology to ensure the reliability of disk [16, 17, 18, 19]. Disk reliability model can be calculated by the data (DT), threshold (TH) and attributes value (AV) of disk SMART parameters. AV has been set to the maximum normal value as default. TH is the fault limit value set by the manufacturers. We confirm the reliability of disk through comparing the output value of disk reliability model with early-warning threshold.

By using the popularity data rank at the present time, we use Zipf-like distribution technology to dynamically predict popularity data in the future time [15], allowing us to get the utilization of disk at the present time and new popularity data rank in the future time. Object-based storage [13] and attribute management [14] technologies have appeared which have the potential to manage popularity data. Attribute management technology realizes management target by user definition; Object-based storage technology can effectively communicate these attributes among different levels. In an object-based storage system, objects are used instead of typical files, which have richer semantic contents and can transfer more information for popularity data. Combined with attribute management technique and through analyzing the users requirements and data access patterns, we can extract the attributes on popularity data, which can makes an implementation of the adaptive management and an improvements on data reliability in storage systems.

When the data migration module has launched, RARM system will open a back end migration thread, which will construct requests and send them to the read-write module of the disk, then reads the data of the source disk and writes it to the target disk. Once the operation is completed, the function returns and the thread can continue its work. RARM will hand up the migration thread and prioritize user requests if the system is busying, and the below process can alleviate the negative impact of the data migration operation on the front end. This allows the process to alleviate the influence of the data migration to the front end.

III. RARM System Software Structure

RARM system is divided into five main software modules: system control, receiver, cache realization, RAID realization, active data repair and migration respectively as shown in Fig. 5 [31].

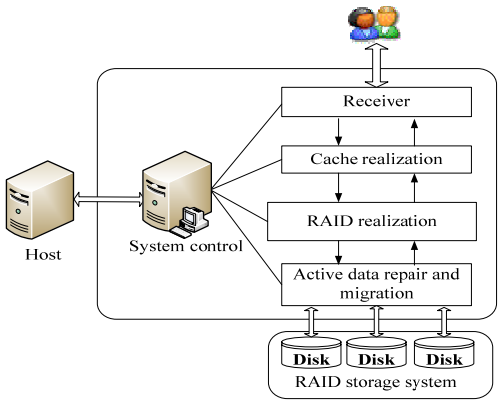


Figure 5. RARM software structure.

(1) System control module manages the whole system and configuration information of the RARM, including

disks, RAID groups (RAID5 [6]) and logical volumes of RAID group information.

(2) The receiver module is responsible for receiving the I/O requests from users, and then transmitting data besides communicating with the fiber-optic equipment, which is also the window of the whole RARM.

(3) Cache realization module is mainly responsible for managing cache of the system, which can speed the access to logical volumes, especially cache those segments of a program that are most frequently used, so this module can accelerate access speed virtually.

(4) The RAID realization module can select appropriate algorithm according to different RAID levels, the module decomposes the user requests and calculates redundant information, then stores it to the specified disks of the RAID groups, while the reading process just combines data block into a whole request and sends to the front user. This module is the key of the RAID system.

(5) Active data repair and migration module is located between RAID kernel program and Linux operating systems. Linux operating system provides the read-write disk interfaces and RAID kernel program trigger active data repair and migration module for provides the read-write disk interfaces. It provides the read-write interfaces for the system call of other modules, and then carries on read-write operations to the disk through interfaces provided by operating system. The study of this paper is focused on active data repair and migration module.

IV. RARM System Function Module

Active data repair and migration module is a link between the upper modules and the underlying disk, the goal is to manage "warning message", which are the bad sectors, reserved space, popularity data attribute and disk reliability information. The system will start RARM operations when sectors or disk reliability condition can't satisfy the standard value and combine the popularity data. RARM system function can be divided into six functional modules: disk read-write request filter, attribute management, data repair, data migration, information data gather and configuration interface. The details of this module are shown in Fig. 6.

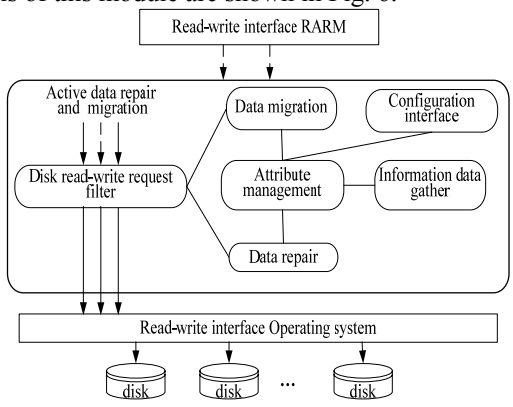


Figure 6. RARM function module division.

Disk read-write request filter is used to analysis implementation status of the read-write requests by intercepting disk I/O, it then reports the information to attribute management. If the requests failed during the read or write process, which failure would indicate that there are some sector failures, the attribute management module will record failure information and call data repair module to repair bad sectors. It may estimate the disk reliable state and predict popularity data at the same time to decide whether or not there is a need to activate data migration to improve data reliability.

When the system is in the process of data migration, there may need to redirect requests to the target disk, the data migration module can make this process through intercepting disk I/O requests [31]. Attribute management module manages disk properties, including popularity data attribute, sectors and disk status and parameters information of other modules. When the current information can't meet the reliability status of standard sector or disk, and system acquires the popularity of data and the utilization of disk, RARM system will dispatch the function modules, and then open a background repair and migration process. Information data gatherer is responsible for start-up data, sector and disk detection at regular time; it provides running state of RARM system at every level, including sector status, disk status, I/O requests status, data status and environmental status. It also obtains system reliability information of different storage device. System design the appropriate WEB page through configuration interface, configure client realize the function of configuration and state-getting through WEB browser, it provides related parameters and state interface to be set by users.

A. Disk Read-write Request Filter

Disk read-write request filter is the driver of RARM, as well as the entry point of all data. The design of this functional module is vital to the portability of the whole modules. In the design, the idea of hierarchical filter disk is used. This functional module provides filter frame and register interface, and other modules can call registered-interface-install filter and filtering disk I/O provided by this module [29]. In the top level of disk I/O filter, disk read-write interface is provided for the rest modules of RAID system. And in the very bottom level of the filter, the operating system's read-write filter is called to execute real disk reading and writing. The structure of disk I/O filter is shown in Fig. 7.

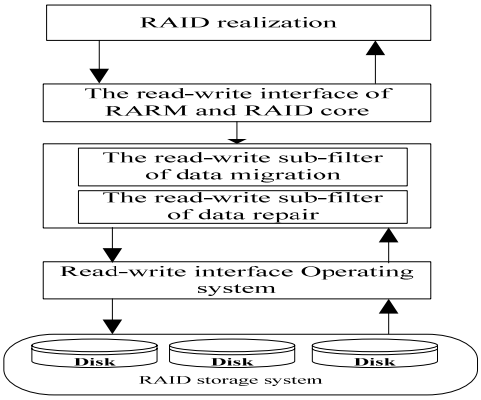


Figure 7. Structure of disk I/O filter functional module.

In Fig. 7, the read-write disk interface provided by operating system mainly calls the interface of SCSI middle layer. SCSI subsystem is in kernel mode, and lies under the file system. SCSI subsystem can be divided into three layers. The top layer is mainly the abstract interface of device type, such as disks, tapes and CD-ROM. And in this layer, no devices' detail can be seen, because it represents the access method of device of a certain kind, and is irrelevant to concrete devices. The middle layer, i.e. SCSI middle layer as mentioned above, is called public layer or uniform layer, which provides a series of general interfaces. The bottom layer is relevant to concrete device driver.

RAID system executes reading or writing disks by calling read-write disk interface provided by RARM. In this time, the disk's I/O request will enter the filter and flow between the sub-filter, and finally go into SCSI middle layer and control the disk to execute real disk read-write operation. When SCSI middle layer completes handling the disk I/O request, it will call the callback function provided by the requester (Here read-write disks all call reading and writing asynchronous interface). The callback function will control the response to return from low to high I/O filter paths, letting each sub-filter gets the response.

B. Attribute Management

Attribute management is the main module of the RARM system module. It controls the data of RAID, and provides interface for calling disk read-write request filter module to acquire the I/O operating condition of storage system. When the disk read-write request filter is done with I/O operation, it will send the I/O operating condition to statistics the wrong times of sector and disk I/O. If the I/O operation is functioning well, the I/O condition will be returned to the disk read-write request filter, or data repair will be used to execute this I/O request again. When the function statistics' wrong times reach a threshold or the information data gatherer detects that the disk is in low reliability status, or when the medium reliability and high utilization disk has popularity data, data migration module will be used to protect data reliability.

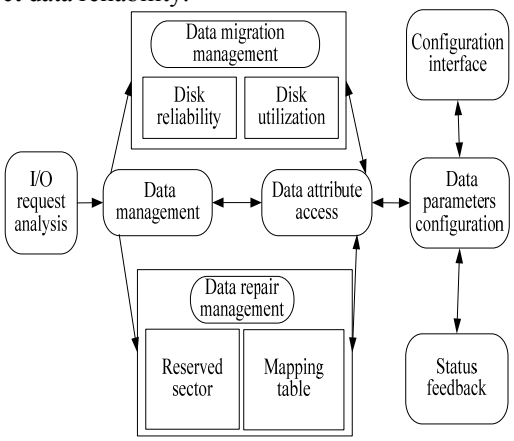


Figure 8. Structure of attribute management functional module.

This module maintains the data of RAID: when the type of data changes, related data will be stored in a given position to prohibit the loss of data attribute. Meanwhile, it provides interactive interface, aiming at the configuration of data parameters, and the feedback of status data. Status feedback includes the data condition of RAID, the repair of data and the schedule of data migration, etc. Fig. 8 presents the plotting picture of attribute management module.

C. Data Repair

Data repair module can effectively shield and replace bad sectors in the disk, including bad sector replacement, data repair and replace mapping table, and the reserved space used by bad sector replacement operations is located by interface calls that the attribute management module provides.

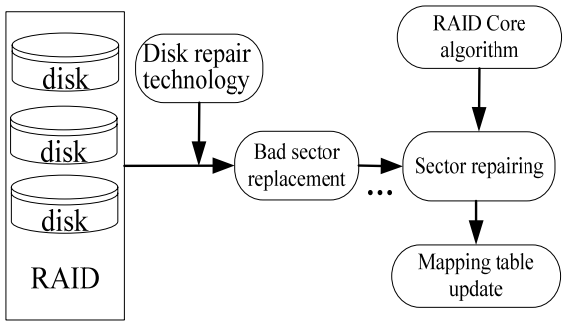


Figure 9. Structure of data repair functional module.

Fig. 9 shows the structure of data repair module. When attribute management module detects disk I/O errors, and leads to bad sectors happen, it will call data repair module uses reserved space to replace bad sectors. Bad sector repairing should incorporate with RAID core algorithm to recover lost sector data. Updating mapping table is the final operation to complete data repair process. It plays a guiding role to the I/O operations of disk, because each disk I/O operations would query updating mapping table to judge whether the sectors of target disk are inside. Data repair module will redirect disk I/O requests to the reserved space if query operations get a positive answer.

Replace mapping table includes source sector and target sector. The structure of the replace mapping table is shown in Fig. 10.

Table item	Source sector LBA	Target sector LBA
	Source sector LBA	Target sector LBA
	...	...
	Source sector LBA	Target sector LBA

Figure 10. Replace mapping table.

As shown in Fig. 10, replace mapping table consists of many table items, each table item further consisting of bad sectors and reserved sectors address. When there is a

bad sector which is replaced to the free sector after researching the reserved sector, it requires adding one item into replace mapping table and describing the corresponding relation of replacement, which is used for the redirection of bad sectors I/O request. Because of added data repair, read-write request requires querying mapping table to make sure whether the requested sector has bad sectors. So the mapping table will be queried often, and it requires better query algorithm, which can decrease performance loss of the query. The query algorithm is designed in the manner of hash table and can enhance query efficiency.

In order to realize data repair technology, RARM reserves a part of space for each disk to replace the bad sectors. Even if the disk has no bad sectors, this space will be reserved also. The size of reserved sector is related to the number of wrong sectors which need to be tolerated, if the reserved space has been used up, the disk will not be able to provide data repair function for a new bad sectors. In RARM system, each disk can reserve 1024*32 sectors, so each disk needs to obligate 16MB space (each sector size is 512 bytes). The maximum number of bad sectors that each disk can tolerate reaches is 20000.

D. Data Migration

When the disk is in low reliability or one disk is medium reliability and high utilization have popularity data, data migration is necessitated to protect the entire data in the low reliability disk or popularity data in the medium reliability and high utilization disk, and the RARM will copy and redirect popularity data of the medium reliability and high utilization disk to the reserved space in high reliability disk or a new one, Data migration uses the disk copy operation, which means copying the data or popularity data of the low reliability disk or medium reliability and high utilization disk in the source drive to a new one or high reliability disk in the target drive. It also uses replace mapping table to confirm whether the target popularity data are inside. The structure of data migration module is shown in Fig. 11.

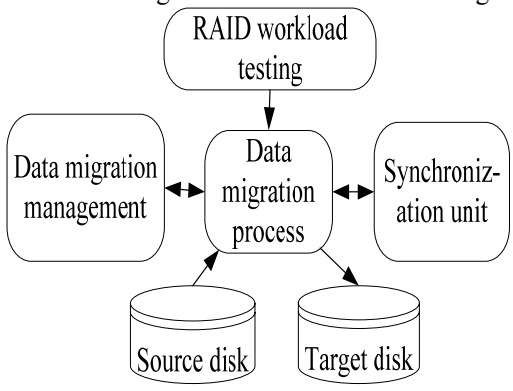


Figure 11. Structure of data migration functional module [31].

In data migration module, background thread for data migration reads data from source disk to system memory, and writes into the target disk. Migration granularity uses a block as the target disk, and executes migration by every 64-128K. Background migration thread reads data from source disk to memory, and if succeeded, writes into the target disk. A data block's migration is completed when the writing operation returns. Then these operations will be done repeatedly, until the whole migration work is completed. During the migration process, it adjusts speed and updates migration management information according to RAID load.

Data migration includes migration schedule, migration bitmap and other information. Migration schedule controls the size of the migrated data, and migration bitmap describes which data blocks in the disk have been migrated successfully. If the migration occurs according to the sequence, migration bitmap needn't be used, because under this circumstance, detailed information about the migration of data block can be known by migration schedule, but disk I/O will be increased when the system is operating well, and using migration bitmap can prevent unnecessary disk I/O [29]. For example, in Fig. 12, three data blocks have been migrated, namely data block A, B and C, migration schedule is three. At this time, the RAID system modifies the fifth data block, namely modify data block E. If there is migration bitmap, we can do writing operation directly with target disk, and update migration bitmap, and reset the count of migrated data block as four. If migration bitmap is lacked, we can only first write data into the source disk before migrating these blocks of data to the target disk according to the order. Obviously, if migration bitmap is lacked, I/O operations brought by migration will be more, especially when RAID system is on heavy load.

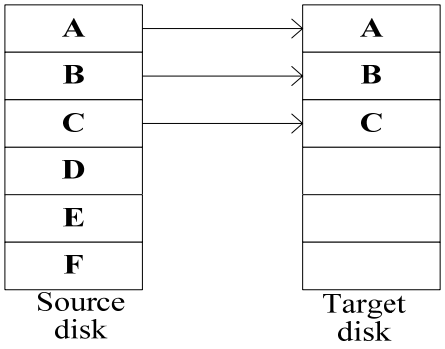


Figure 12. Data block migration.

Migration and actual synchronous writing unit is the synchronization of synchronous migration process and I/O operation to the disk. If synchronization unit is not used, data inconsistency will be caused under certain conditions [29]. Using Fig. 12 as an example to illustrate, data inconsistency will be caused according to the steps below.

（1） the background migration thread reads out data block D from the source disk;

（2） RAID system modifies data block D, and checks migration bitmap to find that this data block is not migrated yet. So it writes the modified data D' into the target disk, and marks it as migrated;

（3） the background migration thread writes the read data block D into the target disk. At these time, data D' is re-modified into D, so the data is not consistent.

Judging from the steps above, if a judgment is added in step 3 about whether a data block has been migrated, it seems to be avoidable, but if during the process step 2 is executed after the judgment, data fault may be caused. The root of this problem is that the migration process is not atomic operation. Reading the source disk operation and writing the target disk operation is not done at the same time. There are several solutions to this problem. All I/O requests can be completed by a single proxy thread. The request, whether put forward by RAID system or the data migration process, will first be put in the handling queue of this proxy thread, and then the process fetches a request to handle. For the I/O request of data migration, the operation of reading source disk and writing target disk is regarded as one operation. In this way the disk I/O atomicity of data migration can be avoided. Besides, the proxy thread can be used to handle the synchronous problem. If the RAID system is functioning upon one single CPU, the effect on the function is rather little. Every I/O operation may include a thread switch, but generally I/O requests are sent in batches. For example RAID has 5 disks. The system reads a stripe every time, and then five I/O requests is thrown in the queue at a time. The system then switches it to proxy thread to handle. In this way the average time consumption of switching decreases. In this paper, synchronization of RAID is realized by proxy thread. The experiments indicate that the influence on speed is not significant. However, if the system is functioning upon multiple CPU, the influence on speed is rather obvious if a proxy thread is created. For instance, in the 4-CPU system, every CPU can take charge of the management of 4 disks (there are 16 disks in all). If a proxy thread is used for disk I/O, one CPU can manage 16 disks, which causes loss of performance (this part is untested because of the condition limit). Fig. 13 shows the structure of writing a request synchronously using a proxy thread. The proxy threads fetch requests from the queue to handle continuously and then handle them one by one. When different modules execute reading and writing operations, the request is put in the I/O request queue. In this way, the request cannot be handled concurrently, so the data inconsistency cannot be caused.

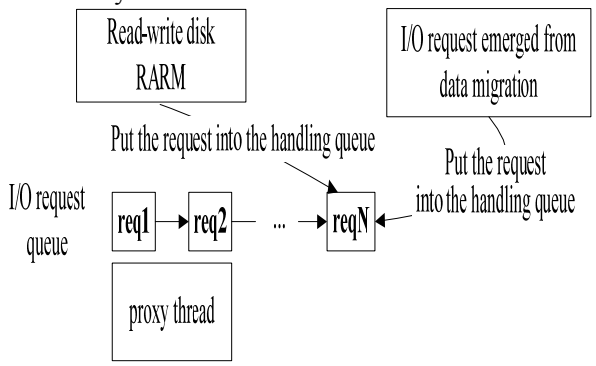


Figure 13. Using proxy thread synchronous RAID system and the I/O request of data migration.

To reduce the effect on system property, the data migration module must adjust migratory speed to system workload dynamically. So there must have a workload testing module. It determines the system workload by counting total number of I/O requests per unit time. Besides, a synchronization unit is also necessary to maintain the data consistency of the migratory data. We design this module by using two components, the timer and the I/O statistical variable. Whenever the I/O requests come, the I/O statistical variable will be updated. After a specified period, we can get system workload information by using I/O statistical variable value compared with the timer value. The module then can inform background processes to adjust the speed of migratory requests. The background processes can take active measures to control the frequency of the migration requests' sending rate, background processes. For instance, the background processes may sleep for a period of time after a certain amount of data is migrated. The measures mentioned above also can improve the property of the front-ends system during data migration.

E. Information Data Gather and Configuration Interface

Information data gather module mainly uses smartmontools tool, attribute management technology, object storage technology, and so on, to detect and acquire of the information data of the RARM system, and provide service for active data repair and data migration. Meanwhile, RARM needs to provide configuration interface to let the users set relevant parameters, which include statistics of disk I/O and so on. Further, status interface is needed to be provided to get the functioning status of RARM, including disk reliability and data migration. Configuration interface module provides these interfaces for external to use, to configure and acquire relevant parameters. As these two modules are not the emphases of this paper, we don't explain in detail here.

V. Evaluation Methodology

In this section, in order to evaluate the system performance's influence based on an active data repair and migration, we evaluate performance loss of increased data repair and data migration based on RAID system using Iometer from the view of the request size, and the performance of prototype implementation of active data repair and migration based on RAID system through extensive trace-driven and benchmark-driven experiments.

A. Experimental Settings

The prototype of RARM system is installed on a storage server that is an iSCSI target. Storage clients are connected to the storage server using the Cisco 3750 Gb Ethernet. The hardware and software details of RARM are listed in Table 1.

TABLE I.
Hardware And Software Of Rarm System.

CPU	Intel iop80321(500MHZ)
RAM	DDR 512MB
DISK	Seagate Barracuda 7200 160G
FC	Agilent 2G
OS	Linux 2.6.11

We use Iometer to run standard benchmarks. Iometer is an I/O subsystem measurement and characterization tool for single and clustered systems especially for storage system. Iometer is based on a client–server model and can run on either linux or windows operating system.

B. Workloads

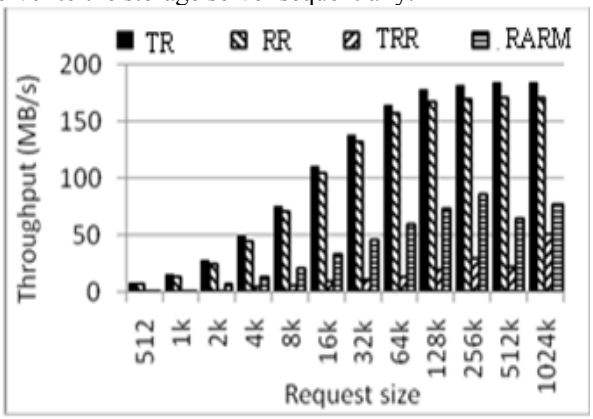
In RARM experiments, we use SPC I/O traces and standard I/O benchmark as our workloads. The traces that drive our experiments consist of three different parts: Financial-1, Financial-2, and Web [8, 9]. The write ratios of these three traces are about 77%, 18% and 0%, respectively. On the client server, we replay the I/O traces using btreplay program of the blktrace tool in Linux. Results of the replay show that I/O requests are generated to the storage server in the form of iSCSI requests. Standard I/O benchmark is discussed above; Iometer can generate synthetically I/O requests [31]. The RAID inside the iSCSI target handles the iSCSI requests.

C. Numerical Results and Discussions

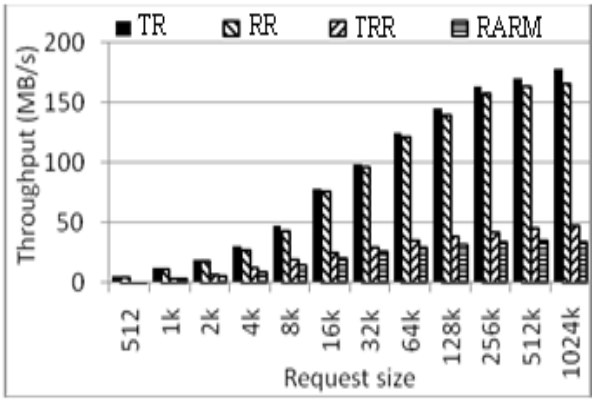
The prototype of RARM system and the experimental settings we use are described in Section V-A. We use RAID5 which has 5 disks in the linux operating system to measure the performance of RARM system. And the volume of the disks we use is 10GB. There are four states of the RAID system: Traditional normal RAID (TR), RAID with data Repair (RR), Traditional Reconstruction RAID (TRR), RAID with Active data Repair and Migration (RARM).

We use Iometer tool to test the throughput of TR, RR, TRR and RARM in different request sizes.

I/O performance: Fig. 14a and Fig. 14b give the results of throughput of the different state of RAID systems, which measure in different request size sent from client server to the storage server sequentially.



(a)Sequential read

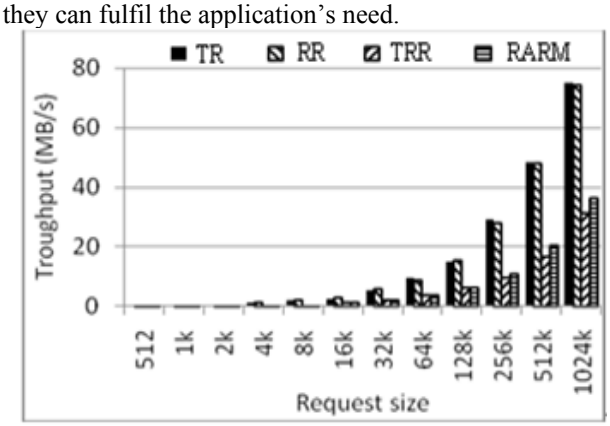


(b) Sequential write

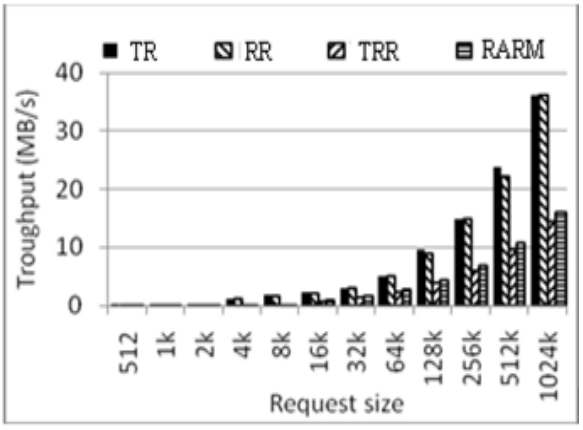
Figure 14. Sequential performance with Iometer.

From the Fig. 14, the throughput of RR is close to TR when the request size from 512B to 1024KB. It means the affection on RAID from the sector's failure is very little in data repair operations. The throughput of RAID system can approach 150MB/s or higher when the request size is set to 256K or above. However the throughput of TRR and RARM can't exceed 90MB/s no matter sequential read or write. But, we also observed that the RARM's throughput is apparently better than TRR in sequential read. The reason is the requests of RARM only need to read source disk or target disk when data is migrating. While TRR needs to calculate the lost data by reading relative data from other disks, which would lose a few times.

Fig. 15 shows the results of random performance. The affection of RARM on system performance is some great because a few embedded filter modules should have more instructions to each I/O operation. But RARM and RR all have good performance in random I/O, so we believe they can fulfil the application's need.



(a)Random read



(b)Random write

Figure 15. Random performance with Iometer.

In Fig. 16, data migration leads to a lot of disk I/O. Because data migration thread works at background, the execution speed is closely related with the CPU occupancy rate. If the I/O speed is too fast, the CPU might become the bottleneck of system performance. We can dynamically adjust data migration velocity to reduce the influence. From Fig. 16 we can see when system I/O workload rises, the data migration velocity adjust automatically to reduce the impact on system performance. Although data migration has some influence on system normally I/O performance, but it is much smaller than TRR system.

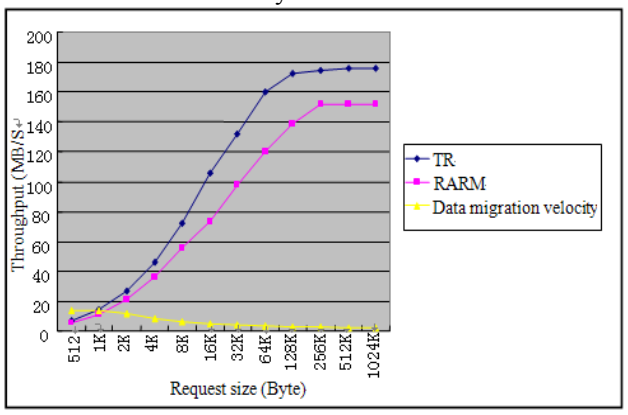


Figure 16. The throughput and data migration velocity of TR and RARM systems with Iometer (RAID5).

In RAID5 there are several disks are working at the same time, it is difficult to test the influence of the disk bottleneck. So we the 2 disks of RAID1 to instead. The result is shown in Fig. 17.

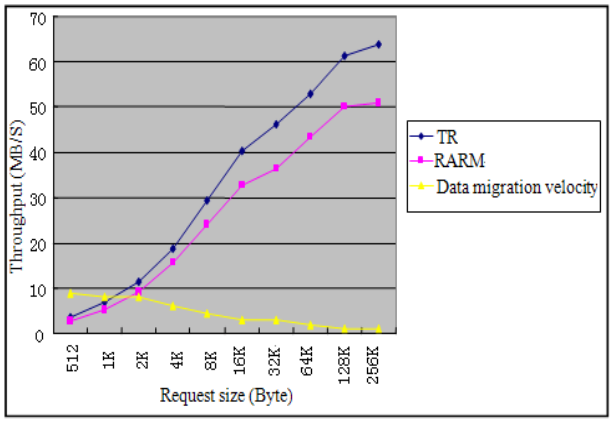


Figure 17. The throughput and data migration velocity of TR and RARM systems with Iometer (RAID1).

We only compare the influence of data migration in sequential read, the situation of sequential write and random read-write is not mentioned here, but they are basically similar. From Fig. 17, the migration speed of RAID1 which has 2 disks is slower than RAID5 which has 5 disks. It is because the disk I/O of RAID5 which has multi-disk will not become a bottleneck, while RAID1 which has 2 disks will be influenced significantly by the disk bandwidth limit when executing read operation.

The practical application results have been given using the SPC traces and benchmarks when disk failure occurs in RAID systems. We use three traces and benchmarks to measure front end performance of average user response time while online recovery operation is in progress and the total online recovery time.

From Fig. 18, the front end performance of RR and RARM on data recovery is excellent. We observed that RR and RARM are 19 and 29 times the average user response time of TRR with trace Financial-1.This is a large improvement of performance. In another word, RR and RARM have very high performance for write-intensive applications, which is shown in Fig. 15(b), but for read only application there is no apparent improvement.

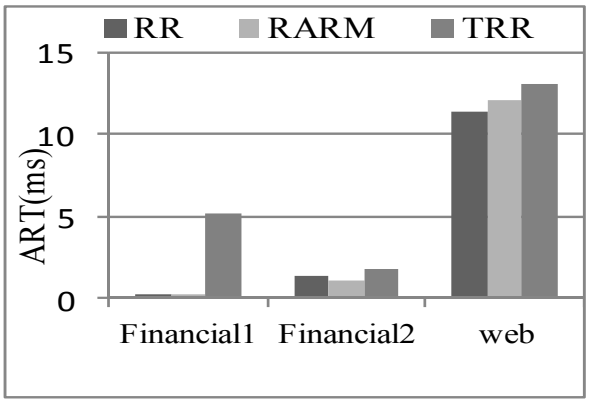


Figure 18. Average response time performance with three traces (ART: ms).

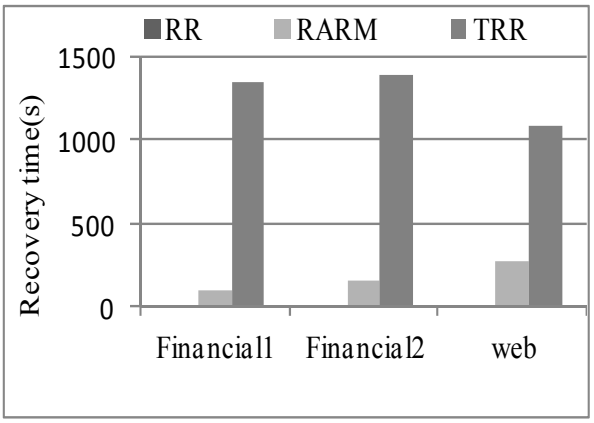


Figure 19. Recovery time performance with three traces (s).

In Fig. 19, the results of RR performance are zero which means the recovery process of RR is very short compared with other two systems. In addition, the operation objects of RR are just the sectors. From Fig. 19, in contrast, the recovery time of RARM decreases rapidly with TRR, the user response time of RARM speeds up significantly, by a factor of up to 13.8 highest and 3.0 lowest. So RARM obviously will be an excellent usefulness for online applications [10] to improve RAID-based storage systems reliability.

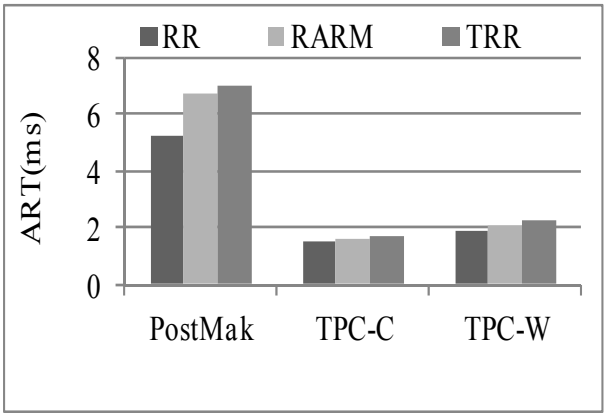


Figure 20. Average response time performance with three benchmarks (ART: ms).

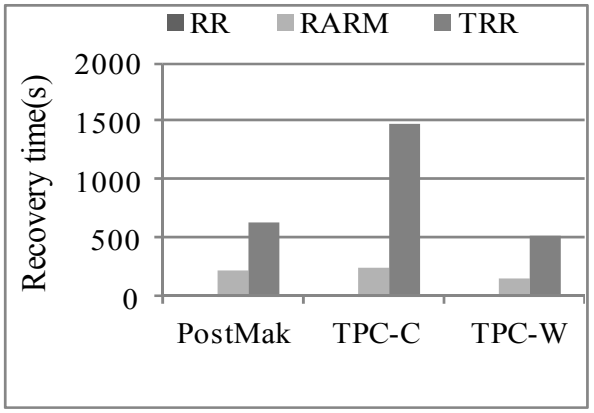


Figure 21. Recovery time performance with three benchmarks (s).

As we can see from Fig. 20 and Fig. 21, the average response time and recovery time of three Benchmark tools are similar to the description of Fig. 18 and Fig. 19, and they have the same results: TRR> RARM >RR.

VI. Related Work

A large number of different approaches for predicting disk based on SMART technology have been studied. Hughes et al [16] proposed Wilcoxon rank sum statistical tests to improve failure warning accuracy and lower false alarm rates. It used Multivariate rank sum along with ORed rank sum and ORed threshold. Murray et al [17] compared the performance of support vector machines (SVMs), unsupervised clustering, and non-parametric statistical tests (rank-sum and reverse arrangements). They also proposed an algorithm based on the multiple-instance learning framework and the naive Bayesian classifier (mi-NB) [19]. Hamerly and Elkan [18] had applied Bayesian methods to predict disk drive failures based on measurements of drive internal conditions. They used two methods. The first method posed it as the problem of anomaly detection. They applied a mixture model of naive Bayes clusters that is trained using expectation-maximization (NBEM). The second method was a naive Bayes classifier, a supervised learning approach. These approaches just predict disk failure, but in our method, the potential predictable sector or disk failures can be predicted. Once the fault has been predicted, the early-warning information should be given immediately, the data in the sector or specified disk should initiatively migrate to other idle.

There are many solutions based on access patterns and workloads in data storage. Wilkes et al [4] utilized hotspot data to improve RAID storage systems. Chervenak et al [20] found that popular movies create hotspots that limit overall performance. Replication of hotspots is cost-effective. Brown et al [21] designed ISTORE, which was a self-maintaining system with recognizing and quenching data hotspots and load imbalance. Hsieh and Kuo [22] discussed the hotspot identification in flash memory storage systems, and proposed a highly efficient method for on-line hotspot identification with limited space requirements. These approaches just predict hotspot data, but in our method, the hotspot data prediction method combines with disk reliability to improve the reliability of hotspot in the medium reliability and high utilization disk.

There has been substantial research reported in the literature on reliability mechanisms in RAID and RAID-structured storage systems. Some of these approaches focus on the improvement of RAID reconstruction algorithms, such as PR [24], PRO [25], WorkOut [26], VDF [27] and others [28]. Our study is related in spirit to active data repair and migration techniques to improve reliability and performance, Our technology can automatically allocate the equivalent space from reserved space and replace the bad sectors, and when the number of bad sectors in the disk exceeds the threshold or one disk is unreliable, the data in the specified disk should initiatively migrate to other idle.

VII. Conclusion

Because the traditional RAID reconstruction has influence on the reliability of storage system, we propose a new RAID system called RARM which uses data repair and migration technologies. We built a RARM in the iSCSI to test the performance of the RAID architecture. Data repair technology can allocate reserved space in every disk and redirect the data from the non-repair sectors to the reserved space. In order to ensure the reliability and utilization of disk, data migration technology copy data from the low reliable disk or the disk which has a few bad sectors exceeding threshold to a new one? It will also copy the popular data of the disk which has medium reliability and high utilization to the reserved space in high reliable disk or to a new disk. In this way, the reconstruction of long data can be avoided.

No matter which technology has been used to protect data, the performance of RARM system such as throughput, average response time and recovery time is better than traditional reconstruct RAID system. The numerical results of SPC traces and standard benchmark show that RARM can improve reliability and performance of the RAID system significantly.

Acknowledgment

This work is sponsored in part by the National Basic Research Program of China (973 Program) under Grant No.2011CB302303 and the National Natural Science Foundation of China under Grant No.60933002, and the HUST Fund under Grant No.2011QN053 and No.2011QN032, and the Fundamental Research Funds for the Central Universities.

References

[1] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," *ACM SIGMOD Record*, vol. 17, pp. 109-116, 1988. doi:10.1145/971701.50214

[2] Q. Xin et al., "Reliability mechanisms for very large storage systems," in *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, San Diego, CA, USA, IEEE, Piscataway, USA, 2003, pp. 146-156. doi: 10.1109/MASS.2003.1194851

[3] J. Y. B. Lee, and J. C. S. Lui, "Automatic recovery from disk failure in continuous-media servers," *IEEE. T. PARALL. DISTR*, vol 13, pp. 499-515, 2002. doi: 10.1109/TPDS.2002.1003860

[4] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan, "The HP AutoRAID hierarchical storage system," *ACM. T. COMPUT. SYST*, vol 14, pp. 108-136, 1996. doi: 10.1145/225535.225539

[5] A. Chervenak, V. Vellanki, and Z. Kurmas, "Protecting file systems: a survey of backup techniques," in *15th IEEE/6th NASA Goddard Conference on Mass Storage Systems and Technologies*, College Park, Maryland, USA, IEEE, Piscataway, USA, 1998, pp. 103-106. doi: 10.1.1.31.7765

[6] P. M. Chen, and E. K. Lee, "Striping in a RAID level 5 disk array," in *1995 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, Ottawa, Canada, ACM, New York, USA, 1995, pp. 136-145. doi: 10.1145/223587.223603

[7] H. H. Kari, H. K. Saikkonen, N. K. Par, and F. Lombardi, "Analysis of repair algorithms for mirrored-disk systems," *IEEE. T. RELIAB*, vol 46, pp. 193-200, 1997. doi:10.110+9/24.589946

[8] UMass Trace Repository, "OLTP Application I/O and Search Engine I/O," http://traces.cs.umass.edu/index.php/storage, 2007.

[9] Storage Performance Council, http://www.storageperformance.org/home.

[10] M. Holland, "On-Line data reconstruction in redundant disk arrays," *Ph.D*, Carnegie Mellon University, USA, 1994.

[11] M. E. Gomez, and V. Santonja, "Characterizing temporal locality in I/O workload," in *2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, San Diego, USA, Society for Modeling & Simulation International, San Diego, USA, 2002.

[12] L. Cherkasova, and M. Gupta, "Analysis of enterprise media server workloads: access patterns, locality, content evolution, and rates of change," *IEEE. ACM. T. NETWORK*, vol 12, pp. 781-794, 2004. doi:10.1109/TNET.2004.836125

[13] M. Mesnier, G. Ganger, and E. Riedel, "Object-based storage," *IEEE. COMMUN. MAG*, vol 41, pp. 84-90, 2003. doi:10.1109/MCOM.2003.1222722

[14] R. Golding, E. Shriver, T. Sullivan, and J. Wilks, "Attribute-managed storage," in *1995 the Workshop on Modeling and Specification of I/O*, San Antonio, TX, USA, 1995. doi:10.1.1.47.5431

[15] C. Wu, X. He, S. Wan, Q. Cao, and C. Xie, "Hotspot prediction and cache in distributed stream-processing storage systems," in *28th International Performance Computing and Communications Conference*, Phoenix, Arizona, USA, IEEE, Piscataway, USA, 2009, pp. 331-340. doi: 10.1109/PCCC.2009.5403810

[16] G. F. Hughes, J. F. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved disk-drive failure warnings," *IEEE. T. RELIAB*, vol 51, pp. 350-357, 2002. doi: 10.1109/TR.2002.802886

[17] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Hard drive failure prediction using non-parametric statistical methods," in *2003 Artificial Neural Networks and Neural Information Processing*, Istanbul, Turkey, MIT, Cambridge, USA, 2003.

[18] G. Hamerly, and C. Elkan, "Bayesian approaches to failure prediction for disk drives," in *Twentieth International Conference on Machine Learning*, Williams College, Williamstown, MA, USA, AAAI, Palo Alto, USA, 2001, pp. 202-209.

[19] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," *J. M. Learn. Re*, vol 6, pp. 783-816, 2005. doi: 10.1.1.84.9557

[20] A. Chervenak, D. Patterson, and R. Katz, "Choosing the best storage system for video service," in *Third ACM International Conference on Multimedia*, San Francisco, CA, USA, ACM, New York, USA, 1995, pp. 109-119. doi: 10.1145/217279.215256

[21] A. Brown, D. Oppenheimer, K. Keeton, R. Thomas, J. Kubiatowicz, D. Patterson, "Istore: introspective storage for data-intensive network services," in *Seventh Workshop on Hot Topics in Operating Systems*, Rio Rico, AZ, USA, IEEE, Piscataway, USA, 1999, pp. 32-37. doi: 10.1109/HOTOS.1999.798374

[22] J. Hsieh, and T. Kuo, "Efficient identification of hot data for flash memory storage systems," *ACM. T. STOR*, vol 2, pp. 22-40, 2006. doi: 10.1145/1138041.1138043

[23] J. G. Wan, J. B. Wang, J. Z. Wang, Z. H. Tan, and M. L. Liu, "RSA: RAID system with Self-healing and Active data migration," in *2010 IEEE International Conference on Intelligent Computing and Intelligent System*, Xiamen, China, IEEE, Piscataway, USA, 2006, pp. 22-40. doi: 10.1109/CICISYS.2010.5658421

[24] J. Y. B. Lee, and J. C. S. Lui, "Automatic Recovery from Disk Failure in Continuous-Media Servers," *IEEE. T. PARALL. DISTR*, vol 13, pp. 499-515, 2002. doi: 10.1109/TPDS.2002.1003860

[25] L. Tian, D. Feng, H. Jiang, K. Zhou, L. Zeng, J. Chen, Z. Wang, and Z. Song, "PRO: A Popularity-Based Multi-Threaded Reconstruction Optimization for RAID-Structured Storage Systems," in *5th USENIX Conference on File and Storage Technologies*, San Jose, CA, USA, USENIX, Berkeley, USA, 2007, pp. 277-290. doi: 10.1.1.118.5269

[26] S. Z. Wu, H. Jiang, D. Feng, L. Tian, and B. Mao, "WorkOut: I/O Workload Outsourcing for Boosting RAID Reconstruction Performance," in *7th USENIX Conference on File and Storage Technologies*, San Jose, CA, USA, USENIX, Berkeley, USA, 2009, pp. 239-252. doi: 10.1.1.145.3886

[27] S. Wan, Q. Cao, J. Huang, S. Li, X. Li, S. Zhan, L. Yu, C. S. Xie, and X. B. He, "Victim Disk First: An Asymmetric Cache to Boost the Performance of Disk Arrays under Faulty Conditions," in *2011 USENIX Annual Technical Conference*, Portland, OR, USA, USENIX, Berkeley, USA, 2011, pp. 173-186. doi: 10.1.1.307.1440

[28] T. Xie, and H. Wang, "MICRO: A Multilevel Caching-Based Reconstruction Optimization for Mobile Storage Systems," *IEEE. T. COMPUT*, vol 57, pp. 1386-1398, 2008. doi:10.1109/TC.2008.76.

[29] Y. Yang, Z. H. Tan, J. G. Wan, C. S. Xie, J. Yu, and J. He, "A reliability optimization method for RAID-structured storage systems based on active data migration," *J. SYST. SOFTWARE*, vol 86, pp. 468-484, 2013. doi:10.1016/j.jss.2012.09.023.

[30] Y. Yang, Z. H. Tan, J. G. Wan, and C. S. Xie, "A reliability optimization method using disk reliability degree and data heat degree," in *7th IEEE International Conference on Networking, Architecture, and Storage*, Xiamen, Fujian, China, IEEE, Piscataway, USA, 2012, pp. 11-22, doi:10.1109/NAS.2012.6.

[31] J. G. Wan, J. B. Wang, J. Z. Wang, Z. H. Tan, and M. L. Liu, RSA: RAID system with Self-healing and Active data migration, in *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Xiamen, Fujian, China, IEEE, Piscataway, USA, 2010, pp. 468-484, doi: 10.1109/ICICISYS.2010.5658421.

[32] T. Yang, B. Shia, J. R. Wei, and K. N. Fang, "Mass Data Analysis and Forecasting Based on Cloud Computing," *Journal of Software*, vol 7, pp. 2189-2195, 2012, doi:10.4304/jsw.7.10.2189-2195.

[33] X. Chen, X. B. He, H. Guo, and Y. X. Wang, "Design and Evaluation of an Online Anomaly Detector for Distributed Storage Systems," *Journal of Software*, vol 6, pp. 2379-2390, 2011, doi:10.4304/jsw.6.12.2379-2390.

[34] B. Liu, S. F. Yang, L. Shi, X. G. Ding and Q. Zhang, "Modeling of Failure Detector Based on Message Delay Prediction Mechanism," *Journal of Software*, vol 6, pp. 1821-1828, 2011, doi:10.4304/jsw.6.9. 1821-1828.

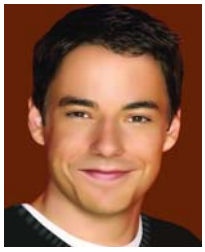
Yin Yang received his Ph.D. in Computer Architecture from the school of computer science and technology at Huazhong University of Science and Technology in 2013. Before joining Wuhan Textile University, he received his Bachelor (2005), and Master (2008) degrees both in Computer Science and Technology from Henan Polytechnic University in China. He is currently working as an Associate Professor at Wuhan Textile University. His research interests include storage security and backup, computer architecture, flash storage, network storage, predict failure analysis.

Wei Liang received her Master in Electronic and Information Engineering from the department of Electronic and Information Engineering at Huazhong University of Science and Technology in 2007. Before joining the 722 Institutes of China Shipbuilding Industry Corporation, she received his Bachelor (2005) from Wuhan Institute of Technology in China. She is currently working as an engineer at the 722 Institutes of China Shipbuilding Industry Corporation. Her research interests include storage reliability, storage security and backup, mobile cloud computing, algorithm analysis.

Zhihu Tan received his Ph.D. in Computer Architecture from the school of computer science and technology at Huazhong University of Science and Technology in 2008. Before joining Huazhong University of Science and Technology, he received his Bachelor (1996), and Master (1999) degrees both in Computer Architecture from Huazhong University of Science and Technology in China. He is currently working as an Associate Professor of the Key Laboratory of Data Storage System at Huazhong University of Science and Technology. His research interests include network storage, storage cache and disk I/O, high availability computing.

Jiguang Wan received his Ph.D. in Computer Architecture from the school of computer science and technology at Huazhong University of Science and Technology in 2007. Before joining Huazhong University of Science and Technology, he received his Bachelor (1996) in Computer Architecture from Zhengzhou University, and Master (2003) degrees in Computer Architecture from Huazhong University of Science and Technology in China. He is currently working as an Associate Professor of the Key Laboratory of Data Storage System at Huazhong University of Science and Technology. His research interests include computer architecture, network storage, high availability computing.

Changsheng Xie is currently working as a Professor and Director of the Key Laboratory of Data Storage System at Huazhong University of Science and Technology, and is also the Associate Director of Wuhan National Laboratory for Optoelectronics. He received his Bachelor (1982), and Master (1999) degrees both in Computer Architecture from Huazhong University of Science and Technology in China. He presented innovative concepts and techniques about Unified Storage Network and Evolution Storage System, and so on. His research interests include ultra-high density optical and magnetic recording, networking storage, storage security and backup, embedded system, high availability computing.



Sandeep Subhoyt is currently a Ph.D. Student in the Computer Science at Virginia Commanwealth University in USA. His research interests include storage reliability, data migration, network storage, predict failure analysis, cloud storage.