Modeling the Impact of Disk Scrubbing on Storage System

Junping Liu, Ke Zhou, Zhikun Wang, Liping Pang, Dan Feng
School of Computer, Huazhong University of Science and Technology
Wuhan National Lab for Optoelectronics, Wuhan, China
E-mail: {liujunping,zkwang}@smail.hust.edu.cn, {k.Zhou, lppang, dfeng }@mail.hust.edu.cn

Abstract—One of characteristics of the rising cloud storage technology is low-cost and high reliability. A distinct benefit of disk scanning or scrubbing operation is identifying the potential failure sectors as early as possible, thus providing high reliability. Obviously, the higher the scrubbing frequency is, the higher the system reliability is. However, it may take a few hours for a scanning process to check the whole disk. In other words, the scrubbing process may result in a downtime or a lower system performance. Furthermore, the scrubbing process consumes energy. In order to reduce the impact of disk scrubbing on disk performance and energy consumption, system designers choose to scan the disk in a low frequency, which results in a lower reliability. Thus it is essential to design a good scrubbing scheme in a large scale storage system over long time horizons. In this paper, we present a novel scrubbing scheme to solve the challenge. In this scheme, an optimum scrubbing cycle is decided by keeping a balance between data loss cost, scrubbing cost, and disk failure rate. Our research shows how the data price and the scrubbing cost affect scrubbing frequency, and the scrubbing scheme is applicable for storage with inexpensive data. Our experiment shows that our scheme outperforms routine method 73.3% in cost and 40% in reliability.

***Index Terms*—Disk Scrubbing, Reliability, System Cost, RAID**

I. Introduction

Data is priceless for enterprises. Therefore, the storage systems which are employed to store the data must guarantee the system reliability and performance[2]. However, disk failures happen frequently in modern data centers in which there are thousands or even tens of thousands of disks. When reliability becomes one of the most key requirements of storage systems, users cannot tolerate those failures. To improve storage reliability, Redundant Array of Independent Disks(RAID) is used in many applications to store huge amounts of data and protect data from disk failures[1]. Whereas, recent research[3][4][7][18] shows that RAID is not foolproof and storage subsystem failures are not independent. After one failure, the probability of additional failures (of the same type) is higher[7]. Thus, there may be an extreme instance in storage system as follows: when we try to rebuild failed data, we discover that the disk containing the block has failed or that a sector in the block can not be read, and then we access the other blocks in the redundancy group in order to recover the failed data, finally, we find that many of the blocks have failed too. In such scenario, we are not able to reconstruct the data which are now lost. System cannot find actively those latent bit errors and it is the reconstruction process that reveals the bit failures in the system.

To avoid this scenario, we must detect or check the potential errors as soon as possible to use the redundancy data built into the storage system. Techniques such as disk scrubbing [1][19][20] and intra-disk redundancy[21] have been proposed to avoid this scenario and maintain high reliability. A widely used scheme is periodically detecting these failures by reading the disks. The detecting operation can check whether the disk failed or not, and verify whether the data stored in the disk are correct or not. If there are sector errors, a repair process is launched to rebuild those failed data. This operation is called “scrubbing” or “scanning”[5].

In general, storage system designers use rules-of-thumb to determine the appropriate solutions to scan or scrub disks. However, every application is unique, and the solution that works well for one application may not be applicable for another one. Thus, improper solution can easily lead to an expensive, suboptimal system approach that fails to meet the users’ requirements.

Based on the above discussion, designing a scrubbing scheme requires to strike a good balance between the costs of the solution and the benefits that the solution can offer. If the scrubbing processes access disks frequently, it will consume more energy and result in a lower system performance. If the disks are accessed less frequently, the probability of block failure is higher and the reliability of system is lower. Therefore, a proper disk scrubbing scheme accounts not only for the needs of the reliability of system but also for the user’s anticipation of performance and cost.

In order to solve this problem, we propose to use a frequency-cost function instead of the deterministic scrubbing[1] cycle or scrubbing frequency to keep a balance between the cost and the reliability. The purpose of this function is to guide storage system disk scrubbing by leveraging the feasibility of cost. The frequency-cost function provides an optimal cost-effective storage solution in terms of the user’s cost/benefit trade-off manner. To the best of our knowledge, no one has used such frequent-cost function to designing the scrubbing scheme in storage system.

The rest of this paper is organized as follows. Related work is discussed in Section 2. We briefly introduce disk

doi:10.4304/jcp.5.11.1629-1637

failure model and scrubbing-cost model in Section 3. Section 4 discusses how to get the optimal disk scrubbing frequency. In section 5, we present the impact of scrubbing parameters on scrubbing frequency and the application field of scrubbing scheme. In section 6, we evaluate the impact of our scheme on the system reliability and the total system cost. We provide some relevant discussions in Section 7. Section 8 summarizes our conclusions and presents future work.

II. Related Work

Disk Scrubbing Techniques: Scrubbing technique was first proposed by Saleh[8]. This technique was used in the memory fault detect. Kari[5] adopted this technique to detect failed disk sectors and implemented three disk scrubbing algorithms. Baker et al.[19] studied threats to long-term data. Using a storage failures model they showed that the most important strategy for increasing the reliability of long-term storage is detecting the latent faults quickly.

The current product in disk scrubbing technology is Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T. Linux)[17] and DISK AEROBICS software[14]. S.M.A.R.T. is a bootable floppy distribution containing tool for removing mechanical and predictable failures in disk drive. DISK AEROBICS software can continuously monitor the disk drive health information and predict the disk failure risk before the failure occurs. Once there is a disk failure risk, it replaces the possible failure disk with a new "healthy" spare drive, thus avoiding lengthy RAID rebuild time and data loss.

In academic doctrines, there are three general issues to study the impact of disk scrubbing on storage system.

The first class research is how disk scrubbing affects the performance of storage system. Iliadis et al.[20] discussed the impact of a scrubbing scheme on the performance of RAID. Their research showed that there is a more degradation in average response time with shorter scrubbing interval and the impact on average response time is negligible when the scrubbing interval is a large scrubbing period.

Bachmat and Schindler[9] analyzed firstly how the low priority disk drive tasks such as disk scrubbing impact the performance of foreground application and then proposed a greedy algorithm to shorten completion times for the low priority disk drive tasks and minimize the negative impact on response times of foreground requests at the same time.

The second class research is how disk scrubbing affects the reliability of storage system. Schwarz et al.[1] analyzed the impact of disk scrubbing on the reliability of storage system by Markov model and proposed a scrubbing scheme which offers an optimal scrubbing interval. Baker et al.[19] and Bairavasundaram et al.[4] showed by analytic model that the most important method for increasing the reliability of archival storage is to find the latent faults quickly.

The third class research is how disk scrubbing affects the energy of storage system. Wang[30] proposed a new metric, energy-reliability product(ERP) to evaluation the impact of disk scrubbing on system reliability.

The above schemes is successful in their original target, however, those schemes didn’t take into account the balance between the scrubbing frequency and system cost which is considered in this paper.

Cost-effective Storage System: There has also been significant research reported in the literature regarding designing cost-effective storage solutions, for example, disaster recovery, which trade off solution costs with expected penalties for data loss and downtime [10][11][12]. Those researches have effectively used utility to trade off the costs of data protection mechanisms against the penalties when data are lost, thus creating minimum (overall) cost solutions for disaster recovery. This result lends support to the notion of using busyness costs as the basis for evaluating storage solutions.

Strunk et al.[6] proposed using utility functions which unifies different system metrics into a single value to evaluate candidate storage configurations and produce an near-optimal configuration. Our scheme also unifies the reliability and the energy consumption into the system cost. Our scheme is close to this approach in this point.

Keeton et al.[16] showed a tool that can provide disaster-tolerant scheme automatically and this scheme aims at financial objectives which are specified by storage administrator.

Our previous work[31] proposed a scrubbing scheme based on cost-effective to design the scrubbing cycle. However, there are two limitations in that scheme. Firstly, it is obviously that the scrubbing cycle of storage system based on RAID5 should differ from the cycle of storage system based on RAID6 because RAID6 has higher reliability than RAID5. Secondly, though it proves that the scrubbing scheme is the optimal in theory, it can’t give any experimental results because of some difficulties[32]. In this paper, we get rid of these limitations.

These tools provide simplistic evaluations of low-level configuration parameters or overall system costs respectively. Neither class of tools can propose a precise and optimum storage system solution to meet user-specified goals.

III. Cost-effective Disk Scrubbing Model

A. Disk Failure Model

Paper[7] shows that the disk failure rates are not constant. Disk drives experience a high failure rate in the early-failure period. The failure rate drops in the first year, also known as infant mortality period. From the second year to the fifth year or even the seventh year, the failure rate will remain relatively constant, and this period is a useful life period. The annual failure rate at this period is approximately 8.8%. At the end of the disks lifetime, the failure rate will rise again, we call this period as a wear out period. In this period, the disk failure rate is very high and we should replace those disks which are experiencing this period. We often refer to this failure model as bathtub model. Especially, the bathtub model can include different failure model that attain by various field data. Storage system administrator can apply our scheme to their storage system according to their actual reliability model.

B. Assumptions

Since systems vary widely, we cannot derive generic prescriptions. In this section, we present some general assumptions that can provide guidance for choosing an optimal scrubbing scheme.

1. The probability distribution function of disk failure moment is F(t), probability density is f(t), and the disk lifetime is T, thus $\mathrm{F(T)}=\int_0^T f(t)dt=1$.

2. When there is a bit error and we don't find this error, the data loss cost that disk run with this failure is directly proportional to the time. The reason is that the longer failure time, the greater the probability of more bit errors, so the great data loss cost. Ratio coefficient Lc is the loss cost of a time unit. The ratio coefficient Lc can be decided by the following formula as discussed in[6]: $Lc = \mathrm{K} \cdot r(t)$. K denotes the data price of per record which is stored in disk or storage system. Generally, K is 202$ per record[27] and the r(t) is the disk failure rate of unit time. When the unit time is per year and the disk failure rate is annual failure time(AFR). For assumption 3, we can know it is not difficult to get the failure rate. We can know from the reliability theory and Schroeder and Gibson's paper[7], the disk failure rate r(t) is defined as: $\mathrm{r(t)} = \dfrac{f(t)}{1-F(t)}$.

3. We assume that the failure time between two successive scrubbing intervals is a uniformly distribution. There are three reasons for this assumption. First, an international organization IDEMA[24] set a standard of disk reliability that the MTBF (Mean Time Between Failure) rating (MTBF can be transformed to failure rate in reliability theory) is a constant in a certain period of disk lifespan. In this standard, the specification of disk reliability is divided into a more detailed MTBF rating, consisting of four different values corresponding to drive ages of 0-3 months, 3-6 months, 6-12 months, and one year to End of Design Life(EODL). Elerath's research[25] also agreed with this standard. Second, Schroeder and Gibson's research[7] showed that the disk replacement rate was in the form of a stair-step. The staircase-like feature shows that the failure rate in a period follows a uniform distribution. The last and also the most important reason is that when the scrubbing frequency is high in a time unit, the scrubbing cycle is a very small value in terms of disk life expectancy. Thus, we may consider that the failure time between two successive scrubbing intervals follows a uniform distribution. We can understand this reason by the following example: the radius of a tree in its top is not the same size as the radius in its root. However, if we cut a little section from the tree, the radius of the section at both ends maybe considered as the same size. To sum up, the failure rate between two successive scrubbing intervals is a uniform distribution.

4. The cost for scrubbing all disks is Sc, the frequency of scrubbing up to the moment T can be expressed as $\int_0^t n(\tau)d\tau$.

5. According to our experiments on RAID reconstruction or scrubbing, when RAID is in the process of reconstruction or scrubbing, the other disk activities are very little. So we can assume that there are no other disk activities when the disk scrubbing is done.

C. Modeling the Cost-Effective Disk Scrubbing

We define the time interval between two successive scrubbings as a scrubbing cycle. We refer to the scrubbing scheme as how to determine the scrubbing cycle. The disk failure time is a random variable that follows an exponential distribution or a Weibull distribution. Once a disk fails, with the assumption that disk will run with failures until the next scrubbing, and this may cause considerable data loss. Obviously, the longer the scrubbing cycle, the greater the loss cost. At the same time, the scanning process may consume power, bring a downtime, and affect the disk performance. The shorter the scrubbing cycle, the more frequent the scrubbing, the higher the cost of scrubbing. Therefore, according to the assumption of a random distribution of failure, the loss cost, and the scrubbing cost, we need a stochastic optimization model to determine the scrubbing cycle, thus minimizing the overall average cost.

Generally, scrubbing cycle is not necessarily constant and should be based on the probability distribution of the failure time. Scrubbing cycle will be shorter when the probability of failure is lower, and vice versa. The probability distribution of disk failure is a continuous probability distribution function. Therefore, the scrubbing cycle is the function of time t, denoted as s(t).. We assume it is a continuous variable. The scrubbing frequency in one time unit is the function of time t, indicated as n(t). Obviously, $n(t) = 1/s(t)$. Generally, the scrubbing cycle is much shorter than the disk uptime, thus we can consider n(t) as a continuous function.

The frequency of scrubbing should have been relative to the time interval, and it is a positive integer. In this model, we consider it as a continuous function of the time T. In this way, we can turn an optimization problem into a functional extreme value problem by adopting mathematical analysis tools.

Assuming a disk keeps running until reaching its lifespan, the objective function of an optimal model is the mathematical expectation of total cost in a running process. If a disk failure is found in the interval $\left[t, t+\Delta t\right]$, according to the assumption 3, the failure time in the cycle s(t) is a uniform distribution, so the mean of uniform distribution is equivalent to the average value of the distribution interval. The time with failure is $s(t)/2 = 1/2n(t)$, loss cost is $\dfrac{\mathrm{L_C}}{\mathrm{2n(t)}}$, and scrubbing cost is $\mathrm{S_{C*}}\int_0^t n(\tau)d\tau$, then, the total cost is $C(n(t)) = \dfrac{\mathrm{L_C}}{\mathrm{2n(t)}} + \mathrm{S_{C*}}\int_0^t n(\tau)d\tau$.

According to the assumptions, we can get a mathematical expectation of the total cost in a running progress as follows:

$$E(C(n(t))) = \int_0^T \left[\frac{\mathrm{L_C}}{2\mathrm{n(t)}} + S_c * \int_0^t n(\tau)d\tau \right] f(t)dt \quad (1)$$

Equation (1) is the frequency-cost function of scrubbing scheme. Though it is possible to quantify the costs and benefits of a specific disk scrubbing solution by leveraging equation (1), it is still a challenge for a system designer to acquire a clear and optimal scrubbing solution due to external constraints.

IV. Optimal Scrubbing Interval

It is implied in the coarse description of disk scrubbing cost above equation (1) that there is a criterion function that depends on scrubbing frequency and cost. Thus, an optimization process for minimizing the criterion function is required. Now we begin to find the optimal scrubbing solutions through the knowledge about calculus of variations as follows.

E(C(n(t))) is the functional of n(t) and it is our objective function. Our objective is to find a n(t) to minimize the C(n(t)). In order to solve the minimum of E(C(n(t))) using Euler formula, it is necessary to change the expression of E(C(n(t))). Let

$$\mathrm{x(t)}= \int_0^t n(\tau)d\tau \quad (2)$$

Obviously, x(t)'s endpoint conditions are as follows:

$$\mathrm{x(t)=0,x(T)\ is\ free} \quad (3)$$

Where x(t) is the frequency of scrubbing till the time t. Thus, equation (1) is changed to

$$E(C(x(t))) = \int_0^T \left[\frac{\mathrm{L_C}}{2\mathrm{n(t)}} + S_c x(t) \right] f(t)dt \quad (4)$$

Equation (4) is an extreme functional problem with one endpoint is fixed and the other endpoint is free.

According to Euler's equation, x(t) should satisfy the following equality:

$$S_c f(t) + \frac{1}{2}\frac{d}{dt}\left(\frac{\mathrm{L_C} \cdot f(t)}{\dot{x}^2(t)} \right) = 0 \quad (5)$$

Compute the integral of equation (5) and note that $\frac{dF}{dx} = \mathrm{f(t)}$, so

$$S_c F(t) + \frac{1}{2}\left(\frac{\mathrm{L_C} \cdot f(t)}{\dot{x}^2(t)} \right) = k \quad (6)$$

Let integral constant k=Sc*a (a is another constant), then

$$\frac{f(t)}{\dot{x}^2(t)} = \frac{2S_c}{L_c}\left[a - F(t)\right] \quad (7)$$

Using the transversal conditions of free endpoint to determine the constant a. According to the relevant knowledge of functional extreme variation:

$$\frac{f(t)}{\dot{x}^2(t)}\Big|_{t=T} = 0 \quad (8)$$

Because F(T)=1, thus a=1.So we can get the following equation (9)

$$\frac{1}{\dot{x}^2(t)} = \frac{2S_c \cdot (1 - F(t))}{\mathrm{L_C} \cdot f(t)} \quad (9)$$

According to assumption 2, equation (2) and x(0)=0, we have

$$n(t) = \sqrt{\frac{\mathrm{L_C} \cdot r(t)}{2S_c}} \quad (10)$$

The scrubbing frequency should obey equation (10) to get the minimal expectation of the total cost. Due to the unrecoverable bit errors rate(UBER) of disks and some RAID can tolerate one disk failure or two disks failure or even more, the coefficient $\mathrm{L_C}$ is different in different storage system that adopt to different RAID level.

If the unrecoverable bit errors rate is considered when we compute the mean time to data loss(MTTDL) of RAID, the probability of unsuccessful RAID reconstruction due to a UBER is as following[29]:

$$P_{recon\_fail} = (N-1) \cdot UBER \cdot Capacity_{disk} \quad (11)$$

N is the number of disks in a RAID group, $\mathrm{Capacity_{disk}}$ is the capacity of disk. The UBER is often easier to get from the disk drive data specifications. Typically, UBER of disk is 10^{-15} for SCSI and 10^{-14} for SATA drives.

The MTTDL of RAID (RAID1, RAID5, etc) which tolerates one disk failure is as following (MTBT denotes the mean time before failure):

$$\mathrm{MTTDL_{RAID} = MTBF / (N \cdot P_{recon\_fail})} \quad (12)$$

The MTTDL of RAID (2-way mirror, RAID6, etc) which tolerates two disks failure is as following(MTTR denotes the mean time to repair):

$$\mathrm{MTTDL_{RAID} = MTBF^2 / (N \cdot (N\text{-}1) \cdot MTTR \cdot P_{recon\_fail})} \quad (13)$$

Supposing $Lc = \mathrm{K} \cdot R(t)$, and $R(t)$ is the failure rate of RAID, K is the coefficient of data loss. We can get the failure rate of RAID from the mean time to data loss (MTTDL), and according to the reliability theory, the failure rate R(t) is 1/MTTDL.

The failure rate of RAID which tolerates one disk failure is(r(t) is the failure rate of disk)

$$R(t)_{RAID} = \frac{1}{MTTDL_{RAID}} = \mathrm{N} \cdot \mathrm{P_{recon\_fail}} \cdot r(t)_{disk} \quad (14)$$

Then, supposing there are M disks in the storage system and M is a multiple of N, the failure rate of storage system is

$$R(t)_{system} = \frac{M / N}{MTTDL_{RAID}} = M \cdot \mathrm{P_{recon\_fail}} \cdot r(t)_{disk} \quad (15)$$

The failure rate of RAID which tolerates two disks failure is

$$R(t)_{RAID} = \frac{1}{MTTDL_{RAID}} = \frac{\mathrm{N \cdot (N\text{-}1) \cdot P_{recon\_fail}} \cdot r(t)^2_{disk}}{\mu} \quad (16)$$

The failure rate of storage system is

$$R(t)_{RAID} = \frac{M / N}{MTTDL_{RAID}} = \frac{M \cdot \mathrm{(N\text{-}1) \cdot P_{recon\_fail}} \cdot r(t)^2_{disk}}{\mu} \quad (17)$$

The scrubbing frequency n(t) when the raid tolerates one disk failure can be expressed like following formula:

$$n(t)_{one\_fail} = \sqrt{\frac{K \cdot M \cdot \mathrm{P_{recon\_fail}} \cdot r(t)^2}{2S_c}} \quad (18)$$

When RAID tolerates two disks failure, the scrubbing frequency can be expressed like following formula (μ is the repair rate and μ =1/MTTR):

$$n(t)_{two\_fail} = \sqrt{\frac{K \cdot M \cdot (N-1) \cdot \mathrm{P_{recon\_fail}} \cdot r(t)^3}{2S_c \cdot \mu}} \quad (19)$$

The formula gives us a reference when a system administrator decides the scrubbing frequency. The scrubbing frequency is varying with the r(t). This result is not applicable for the repair operation performed by administrator. However, we can get the disk failure rate at different disk age from disk specification (Disk failure rate at different disk age are given in some disk specifications), and we can run a process to monitor the disk age and compute the scrubbing frequency through the above formula.

V. The Analysis of Formula (10)

In this section we will discuss some problems about the formula (10).

A. The Value of P_{recon_fail}, μ and r(t)

Supposing that there are 1000 disks in a data center, Each RAID consists of five disks and the storage system has 200 groups of RAID. The capacity of each disk is 120GB, according to the definition of P_{recon_fail}, we can get that P_{recon_fail} = (5-1)*120G*10-14/bit=0.384.

Supposing that we will find the disk failure when there has a disk failure. The reconstruction process is scheduled and the disk bandwidth is approximately 60 MB/s-70 MB/s. According to our experiment on reality RAID, when RAID is on reconstruction mode, the most of disk bandwidth is absorbed by the disk reconstruction process. Thus, we suppose that there are no other IO requests in the RAID reconstruction process. The reconstruction time of RAID which reconstructs the failure disk is $\frac{120GB}{60MB/s \sim 70MB/s} \approx 30$ minutes, that is, MTTR=30 minutes. μ is a ratio and it can not larger than 1, so we can consider that the repair rate μ =1 in one hour.

r(t) is the failure rate of disks. Proverbially, the annual failure rate of disks is varied from 1.7% to 8.6%[18]. In this paper, we consider that the annual failure rate of disks in use-life period is 8.6%.

B. Scrubbing cost

Usually, Sc consists of the loss incurred by the performance degradation and the additional scrubbing power consumption. In this paper, we ignore the loss generated by the performance degradation. There are two reasons for doing so. First, the scrubbing operations are low priority tasks and the impact of scrubbing on performance is negligible. Second, it is difficult to weigh the impact of scrubbing on the system brought by the performance degradation. Thus, we adopt the scrubbing power consumption as the scrubbing cost.

C. The Impact of Parameter K on Scrubbing Frequency

When RAID only can tolerate one disk failure, the reliability of RAID is the same to the reliability of RAID0. Fig.1 (a) and (b) confirms that the scrubbing frequency varies with the disk failure rate associated with the disk age on the condition that the scrubbing cost is a constant.

Generally, the power price is 7.5-8 ¢ per kw·h, and the disk bandwidth is approximately 60 MB/S-70 MB/S, so it takes approximately 120000/60*60*60=5/9 hour (On the assumption that the disk read speed is 60 MB/S) to scrub a 120G disk. The power of a 120 G disk is approximately 13 W (0.013KW). Supposing that there are 1000 disks in a data center, the cost (power consumption) Sc for scrubbing all the disks is 0.013*0.075*1000*5/9=1.625/3$ (Supposing the power price is 7.5 ¢ per kW·h).Therefore, in this section, we consider that the scrubbing cost per time is 1.625/3 $. It must be so specified that the actual power price may be twice the above price in some large cities such as New York, Tokyo, and London etc. Additionally, scrubbing disk also bring new problem, for example, performance degrade[9], extra reliability loss[1][20]. Thus, the value of scrubbing cost should greater than the value in our paper.

We suppose that the disk failure rate follows the bathtub curve discussed in section 3. We vary the loss cost between 0$ to 200$[27]. Like the bathtub curve, the 3D surface is also a bathtub surface. In the bathtub curve, the curve has a sharp decline when the disk age is in infant mortality. Fig.1 (a) and (b) also show that the scrubbing frequency n(t) increases in the disk infant mortality period with the increased data price. In the useful life period, the failure rate r(t) is a constant, but the n(t) is varying with the increase of cost loss. Obviously, the scrubbing frequency is increasing in the wear out period because the failure rate is increasing. The rate of increase in scrubbing frequency is not the same to the rate of failure, and it is constrained by the loss cost. Additionally, in wear out period, the scrubbing frequency should not be the system administrator's key topic, the key topic should be how and when to replace the disks which are in wear out period.

When RAID can tolerate two disks failure, we find that the scrubbing frequency of this storage system is greater than the storage system that tolerates one disk failure in some fixed-interval. We can get the answer from formula (20)

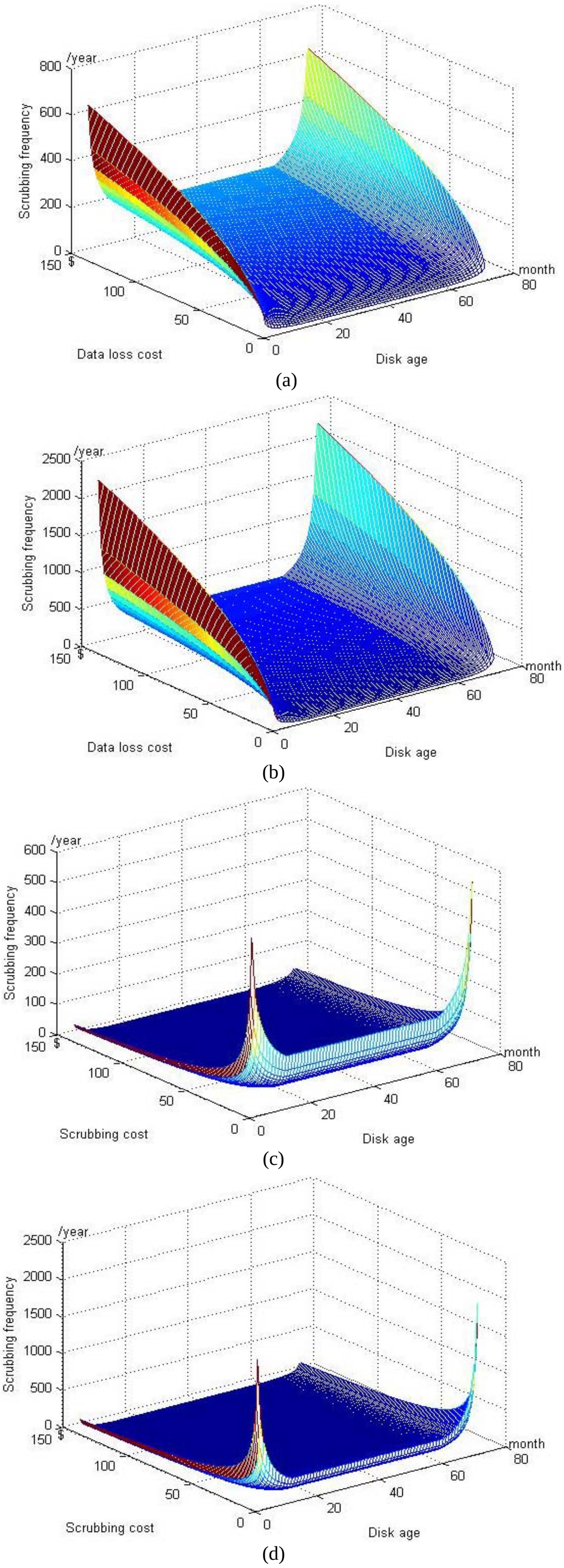
$$\eta = \frac{n(t)_{one\_fail}}{n(t)_{two\_fail}} = \sqrt{\frac{\mu}{(N-1)\cdot r(t)}} \qquad (20)$$

N is a constant and r(t) changes in a certain range, When $(N-1)\cdot r(t) > \mu$, the scrubbing frequency of storage system which tolerates one disk failure is greater than the scrubbing frequency of storage system which tolerates two disks failure.

D. The Impact of Parameter Sc on Scrubbing Frequency

If the user of storage system adapts to different evaluation criteria on scrubbing cost, we could find the impact of scrubbing cost on the scrubbing frequency in this section.

When the loss cost is a constant, the impact of scrubbing cost on scrubbing frequency is shown by Fig 1 (c) and (d). In the early of infant mortality, if the scrubbing cost is low, then we can scrub the disk at a higher frequency. With the increase of the scrubbing cost,



(a)

(b)

(c)

(d)

Figure.1: (a) scrubbing cost is constant and RAID tolerates one disk failure (b) scrubbing cost is constant and RAID tolerates two disks failure (c) loss cost is constant and RAID tolerates one disk failure (d) loss cost is constant and RAID tolerates two disks failure

the scrubbing frequency should be decreased. In the wear out period, we should replace the disks which are extremely unreliable. Fig.1 (c) and (d) show that the scrubbing scheme is not suitable for the storage system consisting of high-capacity disks. The reason is listed as follows: First, the high-capacity disks result in long scrubbing time and more scrubbing cost. Second, the longer time scrubbing brings negative effect to the performance of front-end applications. Finally, according to the researches [22][23], the average use of disk space is approximately 50%. There is a significant waste of resource to scrubbing free space.

When the data price stored in RAID is very costly, the scrubbing frequency will be egregious and impractical. We should adapt another method to protect the security of data. For example, more mirror data which will result a small failure rate or more modern disk which have smaller bit failure rate and disk failure rate.

E. *The Reference Value of Scrubbing Frequency*

The above sections show the relations between the disks age, scrubbing cost and the scrubbing frequency. Table I and Table II show the scrubbing frequency when the data loss cost and the disk failure rate are some typical values.

In Schwarz's paper[1], the optimal scrubbing is 3 times per year. Their scheme supposes that the disk scrubbing is used by storage system such as MAID of which the disks often have a long idle period. In those storage systems, disk will be powered on or powered off to scrubbing. However, this scheme may be not the optimal for a storage system in which the data is accessed frequently. According to the papers[4][20],the scrubbing process scans the entire disk at least once every two weeks. In other words, a realistic scrubbing frequency is at least 24 times per year. As showed in Table I and Table II, some scrubbing frequencies of our results are near to those scrubbing frequencies. Table I illustrates that the scrubbing scheme is applicable to the storage system in which the price of preserved data is mezzo. Once the data is very expensive, the storage system must ceaselessly scan all the disks because overfull scrubbing results in worse system performance[9] and reliability[1]. So the storage system with extremely expensive data should adopt the other reliability mechanism. Apparently, if the data are inexpensive, it is low efficiency to provide excessively reliability.

VI. Experimental evaluation

In this section, we evaluate the impact of our scrubbing scheme on the system cost and reliability by simulation experiment. We compared the cost and the reliability of storage system under various disk scrubbing schemes.

Additionally, all assumptions about the disk failure distribution above in this paper are that the distribution doesn't follow some particular distribution. However, so far as our evidence goes, the evaluation on reliability of storage system all bases on the assumption that the disk failure follows a well-known distribution such as exponential distribution or Weibull distribution. Otherwise, it is impossible that do simulation experiment to study the impact of reliability using existing

Table I: THE SCRUBBING FREQUENCY ON DIFFERENT DATA PRICE AND DISK FAILURE RATE (TOLERATE ONE DISK FAILURE)(YEAR-1)

r(t)	Data Price($)					
	1	10	100	200	300	1000
1.7%	0.32	1.01	3.20	4.53	5.54	10.12
2.0%	0.38	1.19	3.77	5.33	6.52	11.91
2.5%	0.47	1.49	4.71	6.66	8.15	14.88
3.0%	0.56	1.79	5.65	7.99	9.78	17.86
3.5%	0.66	2.08	6.59	9.32	11.41	20.84
4.0%	0.75	2.38	7.53	10.65	13.04	23.81
4.5%	0.85	2.68	8.47	11.98	14.67	26.79
5.0%	0.94	2.98	9.41	13.31	16.30	29.77
5.5%	1.04	3.27	10.35	14.64	17.94	32.75
6.0%	1.13	3.57	11.30	15.98	19.57	35.72
6.5%	1.22	3.87	12.24	17.31	21.20	38.70
7.0%	1.32	4.17	13.18	18.64	22.83	41.68
7.5%	1.41	4.47	14.12	19.97	24.46	44.65
8.0%	1.51	4.76	15.06	21.30	26.09	47.63
8.5%	1.60	5.06	16.00	22.63	27.72	50.61

Table II: THE SCRUBBING FREQUENCY ON DIFFERENT DATA PRICE AND DISK FAILURE RATE (TOLERATE TWO DISK FAILURE)(YEAR-1)

r(t)	Data Price($)					
	1	10	100	200	300	1000
1.7%	0.08	0.26	0.83	1.18	1.45	2.64
2.0%	0.11	0.34	1.07	1.51	1.84	3.37
2.5%	0.15	0.47	1.49	2.10	2.58	4.71
3.0%	0.20	0.62	1.96	2.77	3.39	6.19
3.5%	0.25	0.78	2.47	3.49	4.27	7.80
4.0%	0.30	0.95	3.01	4.26	5.22	9.53
4.5%	0.36	1.14	3.59	5.08	6.23	11.37
5.0%	0.42	1.33	4.21	5.95	7.29	13.31
5.5%	0.49	1.54	4.86	6.87	8.41	15.36
6.0%	0.55	1.75	5.53	7.83	9.59	17.50
6.5%	0.62	1.97	6.24	8.82	10.81	19.73
7.0%	0.70	2.21	6.97	9.86	12.08	22.05
7.5%	0.77	2.45	7.73	10.94	13.40	24.46
8.0%	0.85	2.69	8.52	12.05	14.76	26.94
8.5%	0.93	2.95	9.33	13.20	16.16	29.51

mathematics models. So we think that the disk failure follows Weibull distribution in our experiment. We chose Weibull distribution but not exponential distribution as our experimental distribution based on the following reasons. Firstly, some researches[4][7][18] shows that the disk failure distribution doesn't follow exponential distribution. Secondly, Elerath and Pecht[31] show that the disk failures follow Weibull distribution by field data and attains some distribution parameters. Finally, it doesn't conflict with the bathtub curve which is obtained by actual observation data because the character of Weibull distribution. The character is that the Weibull curve is near to the back of bathtub curve when the shape parameter is greater than 1.In our simulation experiment, the shape parameter is 1.12 and the scale parameter is 0[31].

A. *Cost Analysis*

The aim of our scrubbing scheme is to attain the minimal cost in reliability and scrubbing cost. The simulation experiment results in Fig.2 show that our scrubbing scheme achieves its aims.

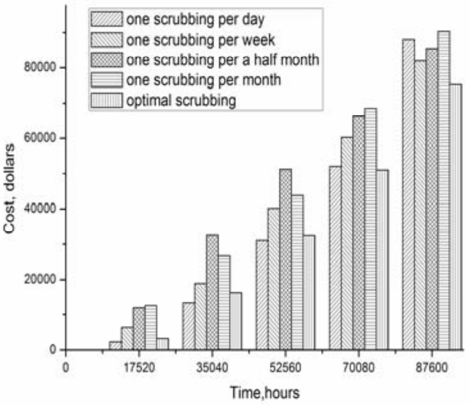


Figure 2: System cost of scrubbing with different scrubbing schemes

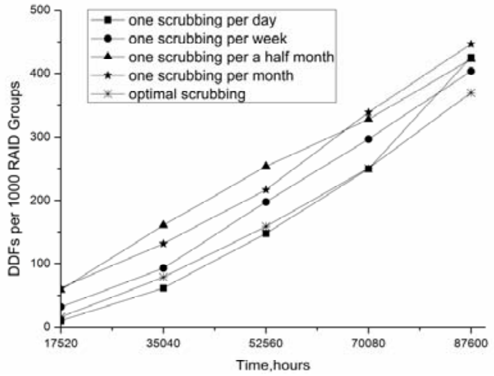


Figure 3: Effects of scrubbing schemes on system reliability

Fig.2 shows that system cost of every scrubbing scheme is increasing with the increase of time. The cost of optimal scrubbing scheme is near to cost of scrubbing scheme that scrubs the disk per day in the early 6 years. With the increase of time, the optimal scrubbing scheme has the lowest cost in those scrubbing scheme and shows its gains. Compared with the scrubbing cycle that scrubs the disk per month, the largest decreasing cost may up to 73.3 percent when the simulation time is 17520 hours.

In Fig.2, at 87600 hours point, the cost of "one scrubbing per day" is higher than the other schemes except the optimal scheme. The reason is relevant to our experiment method. In our simulation experiment, if there are two or more disk failures in a RAID, we think there is data loss. If there are five disk failures in a month in a RAID, when the scrubbing cycle is a day, there are twice data loss. When the scrubbing cycle is a month, there is one data loss. When the reliability of disk is very high (for example, in the middle life of disk), the different caused by this method is ignorable.

Though the optimal scrubbing scheme gets the lowest cost comparing with the other scrubbing schemes, it is doubtful that the optimal scrubbing scheme can provide enough reliability to the storage system. So we do another simulation experiment to evaluate the reliability of system with different scrubbing schemes.

B. *Reliability Analysis*

In this section, we evaluate the impact of scrubbing scheme on system reliability. RAID5 can tolerate one disk failure, and when there are two or more disk failures, data loss will occur. Thus we take the number of double disk failures (DDFs) as the reliability metric in this experiment.

Fig.3 shows that storage system with the optimal scrubbing scheme has the lowest DDFs in all scrubbing schemes. The optimal scrubbing scheme decreases failure times by up to 40%. When the simulation time is smaller than 70086 hours, the system reliability with optimal scrubbing scheme is near to the reliability of system with one scrubbing per day. This phenomenon gives us two lessons. One is that more frequent scrubbing does not necessarily mean that the storage system is more reliable when the system reliability is high. One reason is can get from the part A of this section, the other reason is that extra scrubbing I/O request also result in lower reliability[20]. The other lesson is that data price is the important factor when we select a scrubbing frequency.

VII. Discussions

This paper does not assess the effectiveness of a scrubbing process on the performance of the corresponding storage systems for two reasons. First, for the low-priority scrubbing process, the impact of scrubbing on system performance is slight and the impact can be neglected. Second, for the scrubbing processes which have the same priority as other processes, it is easy to work out the impact on the storage system performance[20].

The disk scrubbing scheme has been used in some disk arrays product, such as the Symmetrix family of disk arrays. The scrubbing cycle in Symmetrix family is a constant rate. However, for any storage system that adopts disk scrubbing scheme, it is apparent that the users will spend more and more money in scrubbing for better reliability as the time increases. Our model also proves this point. However, the scrubbing cost in each operation is much less than the data loss cost, users should adopt scrubbing scheme in their storage systems.

Formula (1) is based on the assumption that user do scrubbing when the disk is idle, and formula (1) couldn't model well when user should launch a scrubbing process. It should do in disk idle period or active period? Our suggestion is that, in different storage system, it should be dealt in different ways. For the archival storage systems such as MAID, Pergamum[28], the optimal scheme to scan the disks is to scrubbing the disks when the disk is powered or busy. The reasons are as follows: firstly, the read/write performance is not the main specification of those systems. It wouldn't increase the scrubbing cost that system adopts the scheme scrubbing the disks when disk is powered or busy. In this case, the model is not necessary to consider the impact of scrubbing scheme on scrubbing cost Sc. Secondly, data storied in these storage systems are rarely accessed and disks in these storage systems often remain powered off between accesses. Powering a disk on and off has a significant impact on the reliability of the disk. A report published by Seagate showed that the MTBF value needs to be multiplied by a factor which related to Power On Hours (POH) of the disk[26]. Thus, the more power on/off cycles is, the lower the reliability of disk is. Thus, the aim that scrubbing disk when disk is actively is to avoid additional power on cycles.

For those storage systems that have a high requirement on performance, the data are accessed frequently, the idle period is likely very short, and it is very frequently that scrubbing active disk (SAD). SAD will increase the scrubbing cost Sc, because SAD has a negative impact on the performance of storage system, and the negative impact results a higher scrubbing cost Sc. There has been no study on the relation between the performance and system cost, and we can't quantitatively study how SAD impact scrubbing cost. However, the formula (10) is still useful, the reason is as follows: first, like the cost of scrubbing operation, the cost brought by the degradation of performance (Dc) is a constant value in a storage system. Dc has no impact on whether the mathematical deduction course from formula (1) to formula (10) is correct. Second, if we must consider the impact of SAD on Dc, users just need subtract a constant value from n(t), and the constant value is decided by the importance of performance.

Additionally, when the disk failure rate follows a certain distribution like those assumptions in other paper, we can analyze the reliability of storage system using the theory of Markov process. In those assumptions, r(t) has a fixed expression. Especially, if the disk failure rate follows index distribution, the disk failure rate will be a constant, then the scrubbing period is decided by the ratio between the data loss cost Lc and the scrubbing cost Sc and the scrubbing cycle is also a constant according to equation (10).

VIII. Conclusions and Future work

Low-cost disk drives with higher capacity but lower reliability are more and more popular in data center. It leads to more frequent data loss coming from sector errors. The scrubbing scheme is adopted to resolve this problem. It needs to balance two competing system metrics, i.e. cost and reliability, in order to choose a proper disk scrubbing scheme.

In this paper, we studied the impact of disk scrubbing frequency on system cost by using an analytical mathematical model. Our research has shown that scrubbing frequency is constrained by data loss cost, scrubbing cost, and disk failure rate. The scrubbing scheme is applicable for storage with low-capacity disk and inexpensive data.

Scrubbing frequency could be influenced by other factors, such as system performance and system energy consumption. Therefore, our future work is to find a global optimum solution which takes into account all these factors. Additionally, we would investigate the disk failure log, conclude the disk failure distribute, and analyze the reliability of storage system using the theory of non-Markov process in reliability theory.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in reviewing this paper. This work was supported by the National High Technology Research and Development Program ("863" Program) of China under Grant 2009AA01A402.

REFERENCES

[1] Thomas J. E. Schwarz, S.J. Qin Xin, Ethan L. Miller, Darrell D.E. Long. Disk Scrubbing Large Archival Storage Systems. In Proceedings of the 12th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2004), October 2004.

[2] Yuhui Deng. Exploiting the Performance Gains of Modern Disk Drives by Enhancing Data Locality. Information Sciences. Vol.179, No.14, 2009, pp.2494-2511.

[3] Weihang Jiang, Chongfeng Hu, Yuanyuan Zhou, and Arkady Kanevsky. Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. The 6th USENIX Conference on File And Storage Technologies (FAST ’08), 2008.

[4] Lakshmi N. Bairavasundaram, Garth R. Goodson, Shankar Pasupathy, Jiri Schindler. An Analysis of Latent Sector Errors in Disk Drives. ACM SIGMETRICS Int’l. Conference on Measurement and Modeling of Computer Systems. Jun. 2007.

[5] H. H. Kari. Latent Sector Faults and Reliability of Disk Arrays. PhD thesis, Helsinki University of Technology, 1997.

[6] John D. Strunk, Eno Thereska, Christos Faloutsos, Gregory R. Ganger. Using Utility to Provision Storage Systems. The 6th USENIX Conference on File And Storage Technologies (FAST ’08), 2008.

[7] Bianca Schroeder, Garth A. Gibson. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? The 5th USENIX Conference on File And Storage Technologies (FAST ’07), 2007.

[8] A. M. Saleh, J. J. Serrano, J. H. Patel, “Reliability of Scrubbing Recovery-Techniques for Memory Systems”, IEEE Transactions on Reliability, vol. 39, no. 1, April 1990, pp. 114-122.

[9] Eitan Bachmat, Jiri Schindler. Analysis of Methods for Scheduling Low Priority Disk Drive Tasks. ACM SIGMETRICS Int’l. Conference on Measurement and Modeling of Computer Systems. Jun. 2002.

[10] E. Anderson, S. Spence, R. Swaminathan, M. Kallahalla, and Q. Wang. Quickly finding near-optimal storage designs. ACM Transactions on Computer Systems, 23(4):337 – 374, November 2005.

[11] J. O. Kephart and W. E. Walsh. An artificial intelligence perspective on autonomic computing policies. In International Workshop on Policies for Distributed Systems and Networks, pages 3 – 12. IEEE, 2004.

[12] M. P. Mesnier, M. Wachs, R. R. Sambasivan, A. Zheng, and G. R. Ganger. Modeling the relative fitness of storage. In Conference on Measurement and Modeling of Computer Systems, pages 37 – 48. ACM Press, 2007.

[13] W. E.Walsh, G. Tesauro, J. O. Kephart, and R. Das. Utility functions in autonomic systems. In International Conference on Autonomic Computing, pages 70 – 77. IEEE, 2004.

[14] http://www.copansystems.com/products/disc_aerobics.php. 2009

[15] Xiaoyan Zhu,Jinhong Cao. The application of bathtub curve in the design and management of reliability. China Quality.7:25-27,2007.(In Chinese)

[16] Kimberly Keeton, Cipriano Santos, Dirk Beyer, Jeffrey Chase, John Wilkes. Designing for disasters. The 2th USENIX Conference on File And Storage Technologies (FAST ’04), 2004.

[17] http://smartlinux.sourceforge.net/. 2009

[18] Eduardo pinheiro,Wolf-Dietrich Weber and Luiz Andr´e Barroso. Failure Trends in a Large Disk Drive Population. The 5th USENIX Conference on File And Storage Technologies (FAST ’07), 2007

[19] Mary Baker, Mehul Shah, David S.H. Rosenthal. A Fresh Look at the Reliability of Long-tserm Digital Storage.In Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys 2006),2006

[20] Ilias Iliadis, Robert Haas, Xiaoyu Hu, and Evangelos Eleftheriou. Disk Scrubbing Versus Intra-Disk Redundancy for High-Reliability RAID storage System. In Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. 2008

[21] A. Dholakia, E. Eleftheriou, X.-Y. Hu, I. Iliadis, J. Menon, and K. Rao. A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors. ACM Transactions on Storage (TOS). Vol. 4, No. 1. May. 2008, pp. 1-42.

[22] John Douceur and William Bolosky, A Large-Scale Study of File-System Contents, ACM SIGMETRICS Performance Review, 59–70, 1999.

[23] Hai Huang, Wanda Hung, and Kang G. Shin, FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption, 18th ACM Symposium on Operating Systems Principles (SOSP), 2005.

[24] The International Disk Drive Equipment & Materials Association (IDEMA). R2-98: Specification of hard disk drive reliability.

[25] J.G.Elerath. Specifying reliability in the disk drive industry: No more MTBF’s. In Proceedings of 2000 Annual Reliability and Maintainability Symposium,pages 194-199.IEEE,2000.

[26] Seagate. Estimating drive reliability in desktop computers and consumer electronic systems.

[27] http://www.compliancebuilding.com/2009/02/09/data-breach-costs-202-per-customer-record/. 2009

[28] Mark W. Storer, Kevin M. Greenan, Ethan L. Miller, Kaladhar Voruganti. Pergamum: Replacing Tape with Energy Efficient,Reliable, Disk-Based Archival Storage. The 6th USENIX Conference on File And Storage Technologies (FAST ’08), 2008.

[29] http://blogs.sun.com/relling/entry/a_story_of_two_mttdl. 2009

[30] Guanying Wang,Ali R.Butt,Chris Gniady. On the Impact of Disk Scrubbing on Energy Savings. Workshop on Power Aware Computing and Systems(HotPower’08),2008.

[31] Jon G.Elerath,Michael Pecht. Enhanced Reliability Modeling of RAID Storage Systems. The 37th Annual IEEE/IFIP International Conference on Dependable System and Networks(DSN’07), 2007.

[32] Junping Liu, Ke Zhou, Zhikun Wang, Liping Pang,Dan Feng. A Novel Cost-Effective Disk Scrubbing Scheme. 2009 Fifth International Joint Conference on INC, IMS and IDC.2009.