Scrub Unleveling: Achieving High Data Reliability at Low Scrubbing Cost

Tianming Jiang, Ping Huang†, Ke Zhou✉
Wuhan National Lab for Optoelectronics, School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, China
†Temple University, Philadelphia, USA
Email: {jiangtianming, k.zhou}@hust.edu.cn, templestorager@temple.edu
Corresponding author: Ke Zhou (k.zhou@hust.edu.cn)

***Abstract*—Nowadays, proactive error prediction, using machine learning methods, has been proposed to improve storage system reliability by increasing the scrubbing rate for drives with higher error rates. Unfortunately, the majority of works incur non-trivial scrubbing cost and ignore the periodic characteristic of scrubbing. In this paper, we aim to make the prediction guided scrubbing more suitable for practical use. In particular, we design a scrub unleveling technique that enforces a lower rate scrubbing to healthy disks and a higher rate scrubbing to disks subject to latent sector errors (LSEs). Moreover, a voting-based method is introduced to ensure prediction accuracy. Experimental results on a real-world field dataset have demonstrated that our proposed approach can achieve lower scrubbing cost together with higher data reliability than traditional fixed-rate scrubbing methods. Compared with the state-of-the-art, our method can achieve the same level of Mean-Time-To-Detection (MTTD) with almost 32% less scrubbing.**

I. Introduction

Although hard disk drives are generally perceived as highly dependable and rarely fail, the devices replaced most frequently in data centers are hard disk drives [1], [2]. Disk storage systems can lose data for a variety of reasons, including failures at both the device level (i.e., complete disk failure [3]) and the block level (i.e., partial disk failure), where individual sectors on a drive cannot be read [4]. As sector errors are latent errors, drive is not aware of and will not report these errors until the affected sector is being accessed, making them challenging to protect against [5]. There are two scenarios where latent sector errors (LSEs) impact data reliability and availability: 1) if LSEs are found while the system operates in degraded mode, e.g., during RAID reconstruction, there will be a data loss; 2) if LSEs are found by a read request, there occurs a temporal data unavailability. Even worse, LSEs happen at a significant frequency in the field. Recent studies based on field data have revealed two drive models among the seven most common hard disk models in a large production installation have 11% and 25% of their drives affected by LSEs, respectively [5].

Nowadays, disk scrubbing and intra-disk redundancy are two common approaches to protect disks against LSEs. Disk scrubbing periodically scans the disks and intra-disk redundancy deploys erasure codes [6], [7]. Though lots of scrubbing schemes [8]–[10] have been proposed, most of them focus on improving reliability without taking cost-efficiency into account. Besides of designing complex scrubbing schemes, there are also efforts in an attempt to predict LSEs using machine learning methods [5]. Mahdisoltani et al. [5] propose increasing the scrubbing rate when errors are alarmed by the prediction results to enhance the data reliability in terms of Mean-Time-To-Detection (MTTD). As shown in Fig. 1, proactive error prediction can further shorten the MTTD.

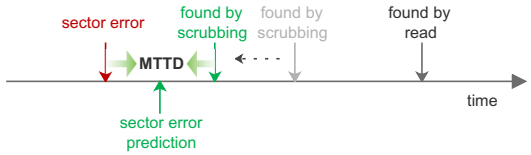


Fig. 1: Both scrubbing and proactive error prediction can shorten the Mean-Time-To-Detection (MTTD).

However, it is very challenging to apply proactive error prediction to guide scrubbing in a practical environment. The first challenge is that scrubbing is not free. Although disk scrubbing enhances the reliability of storage systems, the cost of scrubbing comes in multiple aspects, e.g., energy consumption and impacted performance. It is apparent that the higher rate of scrubbing, the more reliable the storage system will be. On the other hand, higher rate of scrubbing will bring higher cost of scrubbing. However, the budget for data protection is always limited [11], thus making it imperative to build more cost-efficient methods to protect data from being lost. The second challenge is that disk scrubbing is related with not only LSEs but also complete disk failures. As mentioned above, LSEs found while the system operates in degraded mode may incur a data loss, so we need also take the disks with high probability of complete failure into account.

In this paper, we present scrub unleveling, a new approach to improving the cost-efficiency of disk scrubbing using machine learning techniques. Instead of trading the scrubbing cost for data reliability, or vice versa, we aim to achieve low scrubbing cost and high data reliability at the same time.

We make the following three contributions:

- We propose a novel scrub unleveling scheme, using machine learning methods, to adapt the scrubbing rate according to the disks' health statuses. In addition, we collectively study the impacts of complete disk failure and partial disk failure on scrubbing and propose a failure-

978-3-9819263-2-3/DATE19/©2019 EDAA

aware scrubbing scheme.
- Based on the periodic characteristic of scrubbing, we design a novel voting-based algorithm to ensure prediction accuracy.
- We use a real-world dataset to evaluate the performance of our proposed scrub unleveling scheme. Based on the experimental results, by applying scrub unleveling, we can achieve both lower scrubbing cost and higher data reliability than the state-of-the-art method.

II. Theoretical Analysis

In this section, we quantitatively estimate the impacts of the scrubbing rate on the reliability and scrubbing cost. In addition, we also estimate the impacts of introducing error prediction. The symbols used in the rest of this paper are listed in Table I.

TABLE I: Symbols and Definitions

Symbols	Definitions
w	Scrubbing interval
r	Scrubbing rate
X	Factor of scrubbing rate acceleration
Y	Factor of scrubbing rate deceleration
FNR	False Negative Rate
$MTTD_{fixed}$	MTTD under fixed-rate scrubbing scheme
$MTTD_{ada}$	MTTD under adaptive rate scrubbing scheme
M_{factor}	Factor of improvement in MTTD under adaptive rate scrubbing scheme
$Cost_{fixed}$	Scrubbing cost under fixed-rate scrubbing scheme
$Cost_{ada}$	Scrubbing cost under adaptive rate scrubbing scheme
C_{factor}	Factor of improvement in scrubbing cost under adaptive rate scrubbing scheme

To simplify our discussion, we make the following assumptions as in [8], [9], [12].

- The failure time follows a uniform distribution, and the mean of uniform distribution is equivalent to the average value of the distribution interval.
- The impact of normal read operation on MTTD is ignored. As 87% of all latent sector errors in nearline disks are discovered by disk scrubbing [13], the normal read operation's impact on MTTD is negligible.

In standard fixed-rate periodical scrubbing, the scrubber runs continuously at a slow rate in the background as to limit the impact on foreground traffic, i.e., for a scrubbing interval w, a drive is being scrubbed at a rate of $r = 1/w$ [8], [9]. So the average time running with failure is $\frac{w}{2} = \frac{1}{2r}$. Then the MTTD under fixed-rate scrubbing (i.e., without error prediction) is calculated as following:

$$MTTD_{fixed} = \frac{1}{2r} \tag{1}$$

This formulation implies, the higher the scrubbing rate, the higher the reliability. Note that lower MTTD means higher level of reliability.

In scrub unleveling, we accelerate the scrubbing rate by a factor $X(> 1)$ if error is predicted, otherwise decelerate the scrubbing rate by a factor $Y(< 1)$. We use TP and FN to represent the number of true positive and the number of false negative, respectively. The time to detect one true positive (TP) is $\frac{1}{X*2r}$ with accelerated scrubbing rate $X * r$. And the total time to detect all TPs is proportional to the number of true positive disk, so the result is $\frac{1}{X*2r} * TP$. Similarly, the total time to detect all false positives (FPs) is $\frac{1}{Y*2r} * FN$ with decelerated scrubbing rate $Y * r$. Then the MTTD of scrub unleveling scheme (denoted as *adaptive*, and we use them interchangeably in the following discussion) is calculated as following:

$$\begin{aligned} MTTD_{adaptive} &= \frac{\frac{1}{X*2r} * TP + \frac{1}{Y*2r} * FN}{P} \\ &= \frac{1}{X * 2r} * (1 - FNR) + \frac{1}{Y * 2r} * FNR \end{aligned} \tag{2}$$

where $P = TP + FN$ and $FNR = \frac{FN}{P}$. Note that the $MTTD$ is just related to FNR, for only the erroneous disks impact the reliability.

Compared with fixed-rate scrubbing, the factor of improvement in MTTD under adaptive rate scrubbing is calculated as following:

$$\begin{aligned} M_{factor} &= \frac{MTTD_{fixed}}{MTTD_{adaptive}} \\ &= \frac{1}{X} * (1 - FNR) + \frac{1}{Y} * FNR \end{aligned} \tag{3}$$

In a traditional fixed-rate periodic scrubbing, the scrubbing cost is proportional to both the total number of disks and the scrubbing rate, since higher scrubbing rate means more scrubbing overhead. Under the assumption of scrubbing rate being r and total number of disks being $P+N$, the scrubbing cost, within time span T, of standard periodical scrubbing (i.e., without error prediction) is calculated as following:

$$Cost_{fixed} = T * r * (P + N) \tag{4}$$

In scrub unleveling scheme, we apply accelerated scrubbing in response to predicted positive (i.e., PP) and apply decelerated scrubbing rate in response to predicted negative (i.e., PN). Then the scrubbing cost, within time span T, under adaptive scheme is calculated as following:

$$Cost_{adaptive} = T * (X * r * PP + Y * r * PN) \tag{5}$$

where $PP = TP + FP$ and $PN = FN + TN$.

Compared with fixed-rate scrubbing, the factor of increase in scrubbing cost under adaptive rate scrubbing is calculated as following:

$$\begin{aligned} C_{factor} &= \frac{Cost_{adaptive} - Cost_{fixed}}{Cost_{fixed}} \\ &= \frac{X * (PP) + Y * (PN) - (P + N)}{P + N} \\ &= \frac{(X - 1) * (PP) + (Y - 1) * (PN)}{P + N} \end{aligned} \tag{6}$$

where $PP + PN = P + N$.

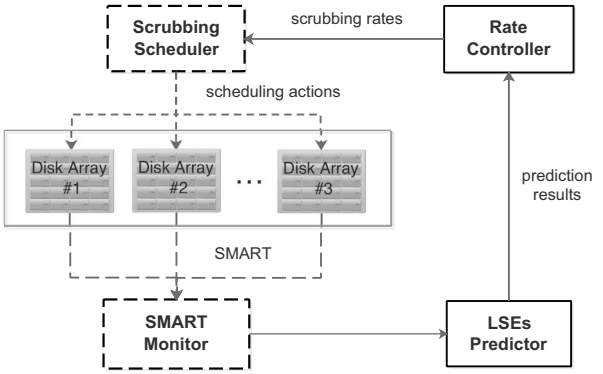


Fig. 2: System architecture of Scrub Unleveling. A **SMART monitor** collects and aggregates the SMART values. The centralized **LSEs predictor** predicts the presence of LSEs and forwards prediction results to **rate controller** which guides the **scrubbing scheduler** via adaptive scrubbing rates.

III. Scrub Unleveling

In scrub unleveling, we adjust the scrubbing rates for each individual disk according to its health statuses. Fig. 2 shows the architecture of scrub unleveling. A monitor collects and aggregates the attribute values reported by Self-Monitoring, Analysis and Reporting Technology (SMART). A centralized LSEs predictor reads the SMART attributes values, calculates the probabilities of LSEs using machine learning methods, and provides notifications about impending LSEs. The rate controller receives the prediction results and guides the scrubbing scheduler via adaptive scrubbing rates.

A. LSEs Predictor

SMART is a monitoring system implemented inside most of modern drives that monitors and reports various indicators of drive reliability [5]. The goal of the LSEs predictor is to predict whether a drive will have LSEs within a given time interval, based on the collected SMART data. We formulate the problem of predicting future errors as a binary classification problem and then use a variety of machine learning methods to train classifiers. As in practice a common rule of thumb is to scan the entire disk drive once every two weeks [5], [8], [13], [14], we attempt to predict LSEs two weeks into the future, i.e., whether there will be LSEs within the next 14 days.

The predictor performs predictions every day, but the scrubbing is issued once every scrubbing period. Given the periodic characteristic of disk scrubbing, we take a voting-based algorithm, shown in Algorithm 1, to combine the results in current scrubbing interval. In detail, when making a prediction for a drive, we check all the N samples in current scrubbing interval w and output erroneous if more than N/2 samples are classified as erroneous, otherwise healthy. Different from the voting-based algorithm used in [15] where the windows for voting are fixed, the windows used in our method are determined by the disk scrubbing intervals, i.e., if the current scrubbing interval is w, the current window for voting is set as w.

Algorithm 1 Voting-based LSEs prediction algorithm

Input: The sample set S[1..t] of the drive, the prediction model Predictor() which returns 0 if the input sample is classified as healthy and 1 otherwise, and the original scrubbing window size w (i.e., the first voter turnout), the accelerated factor $X(>1)$, the decelerated factor $Y(<1)$

Output: healthy or erroneous

initialize $voter[i] \leftarrow 0$ for $i = 1$ to w
$count \leftarrow 0$ ▷ count the number of samples have voted
for $j \leftarrow 1, t$ **do**
  $count \leftarrow count + 1$
  $voter[count] \leftarrow Predictor(S[j])$
  if $count == w$ **then** ▷ current scrubbing is finished
    if $\sum_{i=1}^{w}(voter[i]) > w/2$ **then**
      forward erroneous to rate controller and adjust the scrubbing window w for next scrubbing as:

$$w = w/X \tag{7}$$

    else
      forward healthy to rate controller and adjust the scrubbing window w for next scrubbing as:

$$w = w/Y \tag{8}$$

    end if
    update $voter[i] \leftarrow 0$ for $i = 1$ to w
    $count \leftarrow 0$
  end if
end for

B. Scrubbing Rate Controller

Scrubbing a storage system requires balancing the costs of the scrubbing against the benefits that the scrubbing will provide. In this paper, we define the benefits as data reliability and the costs as energy consumption of scrubbing which is proportional to scrubbing time.

In Section II, we have analyzed the positive impacts brought by error prediction guided scrubbing. Besides of the LSEs, the varying complete disk failure rates also render a constant scrubbing rate to be suboptimal. According to the bathtub model [2], the first year of operation is characterized by early failures (or infant mortality). In years 2-5, the failure rates are approximately in steady state, and then, after years 5-7, wear-out starts to kick in. As a result, scrubbing at a constant rate is not enough in the disk infant mortality period and wear-out period, while it is over-scrubbing in useful-life period. Therefore, we encode the disk failure characteristics into scrubbing scheduling to better protect the data. The detailed scrubbing rate controller algorithm (denoted as adaptive+, which is an improved adaptive scheme) is shown in Algorithm 2, in which the scrubbing rate is adapted by both partial and complete failure probabilities.

IV. Dataset Description and Preprocessing

A. Dataset

To evaluate the proposed schemes, a real-world dataset from Backblaze [16], which spans a period of 12 months from Jan-

Algorithm 2 Scrubbing rate controller algorithm

Input: Scrubbing rate r_1 for useful-life period, scrubbing rate r_2 for infant mortality period and wear-out period, accelerated factor $X(> 1)$, decelerated factor $Y(< 1)$, the power-on hour set $POH[1..N]$ of disks, the health status set $Health[1..N]$ of disks

Output: scrubbing rate r[1..N] of disks

```
for i ← 1, N do
    if POH[i] < 1y || POH[i] ≥ 6y then        ▷ disks in their infant mortality period and wear-out period
        if Health[i] == erroneous then
            r[i] ← X * r2                      ▷ accelerate scrubbing rate
        else
            r[i] ← Y * r2                      ▷ decelerate scrubbing rate
        end if
    else                                       ▷ disks in their useful-life period
        if Health[i] == erroneous then
            r[i] ← X * r1                      ▷ accelerate scrubbing rate
        else
            r[i] ← Y * r1                      ▷ decelerate scrubbing rate
        end if
    end if
end for
```

TABLE II: Overview of dataset

Drive Model	Class	No. Drives	No. Samples
ST4000DM000	no-error	34,830	11,739,000
	error	328	8,693
ST8000DM002	no-error	9,745	3,378,084
	error	223	6,527
ST8000NM0055	no-error	14,299	2,455,296
	error	175	4,583

TABLE III: The 12 selected SMART attributes

Attribute ID	Attribute Name	Attribute Type
1	Real_Read_Error_Rate	Normalized
3	Spin_Up_Time	Normalized
4	Start_Stop_Count	Raw
5	Reallocated_Sector_Count	Raw
7	Seek_Error_Rate	Normalized
9	Power_On_Hours	Normalized
10	Spin_Retry_Count	Normalized
12	Power_cycle_Count	Raw
187	Reported_Uncorrect	Normalized
194	Temperature_Celsius	Normalized
197	Current_Pending_Sector	Raw
198	Offline_Uncorrectable	Raw

uary 2017 to December 2017, is collected. From this dataset, we select three drive models, Seagate's ST4000DM000, ST8000DM002, and ST8000NM0055, which are affected by LSEs most significantly, i.e., the number of drives affected by LSEs is among the most in this period. The definition of a sector error event is defined as in [5]: a drive is considered to have a sector error if its raw value of SMART 5, which indicates the total number of reallocated sectors, increases.

As shown in Table II, erroneous drives indicate drives that have at least one sector error in the year of 2017. For erroneous drives, samples in a period of two weeks before actual sector error are labelled as error, otherwise no-error. For no-error drives, we label the total samples as no-error. It is worthy to note that two weeks is the common scrubbing period used in storage systems [5], [8], [13], [14].

B. Data Preprocessing

1) Imbalanced Dataset: As shown in Table II, the dataset is highly imbalanced, i.e., error disks are much fewer than no-error ones. To address this problem, we employ an under-sampling method, which is commonly used [17], to under sample the majority class, resulting in different ratios of error to no-error samples ranging from 1:1 to 1:50. In the final training set, the ratio of error to no-error samples is set to 1:3 which leads to the best prediction accuracy.

2) Feature Selection: We employ feature selection to remove redundant and irrelevant features, and select relevant features. This preprocessing can not only reduce the time of model training and prediction, but enhance the prediction performance [10]. In our dataset, each disk reports 24 SMART attributes, and each attribute contains a raw value and a normalized value. Instead of factoring in all the SMART attributes values into prediction model, we use correlation coefficients and select 12 features that correlate most with LSEs. The selected SMART attributes are listed in Table III.

3) Normalization: Since different SMART attributes have diverse value intervals, to acquire a fair comparison among them, we apply data normalization. The Z-score normalization used in our approach is calculated as following:

$$x_{normal} = \frac{x - \mu}{\sigma} \tag{9}$$

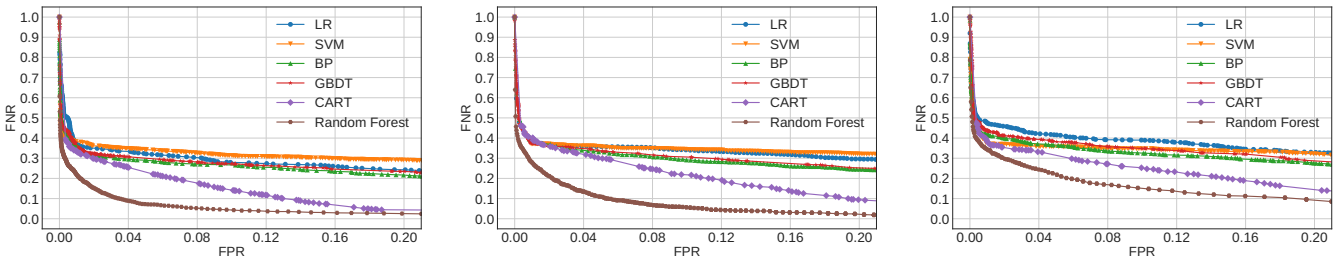
where x is the original value of a feature and μ and σ are the mean and standard deviation of the feature in our dataset, respectively. We also experiment with Min-Max normalization [5], [18], but find Z-score results in a better prediction performance.

V. Experimental Results

A. Results of Proactive Error Prediction

To evaluate the models, we divide the dataset randomly into training and test sets for each drive model. The training set consists of 70% of all error and no-error drives, and the remaining 30% of the drives are in the test set.

To build the LSEs prediction, we have evaluated six different machine learning methods, including logistic regression (LR), random forests, support vector machines (SVM), classification and regression trees (CART), backward propagation neural networks (BP) and Gradient Boosting Decision Tree (GBDT). For LR we experimented with L2 regularization and learning rate of 0.01. For random forests we experimented with different numbers of trees, and settled on using 200 trees for the results included in the paper. For SVM we used the



(a) ST4000DM000. (b) ST8000DM002. (c) ST8000NM0055.

Fig. 3: False positive rates (x-axis) versus false negative rates (y-axis) when predicting sector reallocation (SMART 5).

LIBSVM library [19], and experiment with linear kernel. For BP, we use 3 layers BP with 64 nodes in the hidden layer. Both hidden and output layers use relu function as activation function. We set the maximum number of iterations to 2000, the learning rate to 0.01 and adopt Adam [20] for optimization. For GBDT, we use 100 trees and learning rate of 0.1.

The quality of the predictions is shown in Fig. 3, in which each line corresponds to a different machine learning method. The X-axis shows the FARs which range [0;0.2] and the Y-axis shows the FNRs. As we can see, errors can be predicted with a high enough accuracy to guide lighter-weight proactive actions [5]. E.g., when limiting the FPR to 10% we can correctly predict 96%, 94%, and 85% of all errors for ST4000DM000, ST8000DM002, and ST8000NM0055, respectively. When comparing different machine learning methods, the random forests consistently outperform other classifiers. One possible reason is that random forests have few parameters to tune, making them easily to train. For other classifiers, a considerable amount of tuning is required, making them hardly to achieve reasonable high accuracy [5].

B. Simulating Scrub Unleveling

Based on prediction results, we build simulators to evaluate the effectiveness of our scrub unleveling schemes. We choose the random forests as predictor, as they consistently provide the highest quality predictions. In the adaptive scheme, we set the base scrubbing rate as r_1 in the whole lifetime of disk. In the adaptive+ scheme, we set the base scrubbing rate as r_2 in both the disk infant mortality period and wear-out period, and set the base scrubbing rate as r_1 in useful-life period. In all simulations, we set r_1 to one full disk scrubbing bi-weekly which is a common scrub rate in practice. For the bathtub characteristic of life cycle failure pattern for hard drives, we set $r_2 = 2 * r_1$, i.e., one full disk scrubbing per week.

The choice of the scrubbing rate will depend on the system's sensitivity to added scrubbing cost. To this end, we experiment with a range of X and Y values. For sequential scrubbing is the common approach used in production, we use sequential scrubbing in our simulation. It is worthy to note that our scheme is compatible with other complex scrubbing approaches, e.g., staggered scrubbing [8].

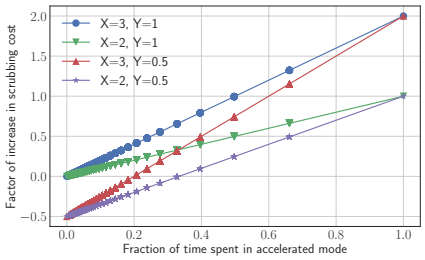
Fig. 4 shows scrubbing cost and MTTD under different accelerated scrubbing modes. Due to space limit, we only show the results of Seagate's ST4000DM000. Each line corresponds to a different factor X of acceleration and factor Y of deceleration. And each point on every single line corresponds to a different pair of (FPR, FNR), i.e., a different prediction result, which determines the proportion of time spent in accelerated mode. It is worthy to note that we either accelerate or decelerate the rate based on the results of LSEs prediction. That is to say, the sum of the proportion of time spent in accelerated mode and that in decelerated mode is 1.

1) Scrubbing Cost: As shown in Fig. 4a, the factor of increase in scrubbing cost is proportional to the time the scrubbing spends in accelerated mode. That is to say, the higher rate of the scrubbing the higher scrubbing cost we will have to pay. Although, low scrubbing cost is our goal, we need also take into account data reliability.

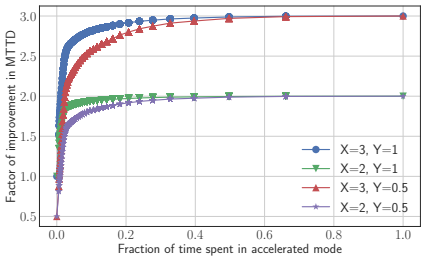
2) MTTD: Fig. 4b shows the factor of improvement in MTTD under different fractions of time spent in accelerated mode. We observe that even if we limit the fraction time the system spends in the accelerated scrubbing mode to 5% (i.e., the storage system spends 95% of the total time in the decelerated scrubbing mode), the factor of improvement in MTTD is still promising.

The inevitable false negatives in the error prediction could cause LSE to be found in a delayed manner, resulting in less improvement of MTTD for the scrubbing rates of these disks are decelerated. So lower deceleration factor Y, with the same acceleration factor X, will lead to less improvement in reliability. However, this sacrifice is worthy, for the healthy drives are the major and the ratio of false negative is very low in our prediction as shown Section V.

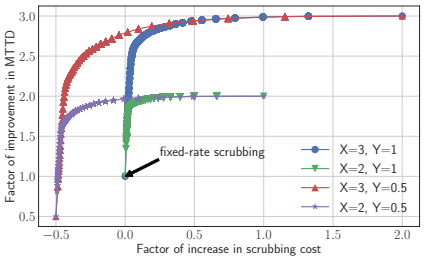
3) Cost Versus MTTD: As shown in Fig. 4c, we can achieve positive increase of MTTD with reduced scrubbing cost when compared with fixed-rate scrubbing indicated by point (0,1). For detailed comparison of adaptive scheme, adaptive+ scheme and the state-of-the-art, we simulate them with the pair of (X, Y) set as (2, 0.5). From the results, shown in Fig. 5, we observe that adaptive+ scheme can still achieve higher improvement in MTTD when extra scrubbing cost is zero. Moreover, compared with fixed-rate scrubbing, we can achieve the same level of reliability with almost 49% scrubbing cost reduced, or improve the reliability of storage systems by a factor of 2.4X in terms of MTTD without extra scrubbing cost. And compared with the state-of-the-art (i.e., proactive error prediction proposed by Mahdisoltani et al. [5]), our method can achieve the same



(a) Scrubbing cost. The X-axis shows the fraction of time the scrubbing spends in accelerated mode and the Y-axis shows the factor of increase in scrubbing cost.



(b) MTTD. The X-axis shows the fraction of time the scrubbing spends in accelerated mode and the Y-axis shows the factor of improvement in MTTD.



(c) Scrubbing cost versus MTTD. The X-axis shows the factor of increase in scrubbing cost and the Y-axis shows the factor of improvement in MTTD.

Fig. 4: Simulating the scrub unleveling. Each line corresponds to a different pair of (X, Y).

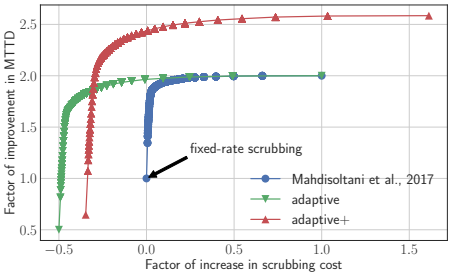


Fig. 5: Adaptive versus adaptive+. The X-axis shows the factor of increase in scrubbing cost and the Y-axis shows the factor of improvement in MTTD. And the pair of (X, Y) is set as (2, 0.5) for all schemes.

level of MTTD with almost 32% scrubbing cost reduced. This is because that the scrubbing resources are allocated more efficiently, i.e., saving unnecessary scrubbing resources on healthy disks to disks at risk.

VI. Conclusion

In storage systems, disk scrubbing has to be performed to maintain high data reliability in response to LSEs. However, the presence and severity of LSEs exhibited by individual disks bear wide variations, leading existing scrubbing schemes to be cost-ineffective. To address this problem, we propose a new scheme, scrub unleveling, to achieve high reliability at low scrubbing cost. By using the results of LSEs prediction, we enforce a lower rate scrubbing to healthy disks and a higher rate scrubbing to disks subject to LSEs. To evaluate the effectiveness of scrub unleveling, we conduct mathematical analysis, simulations and experimental measurement. Experiments with a real-world dataset have demonstrated the superiority of our method over the state-of-the-art for achieving high data reliability at low scrubbing cost.

Acknowledgement

The research is supported by the Innovation Group Project of the National Natural Science Foundation of China, No. 61821003 and the National Key Research and Development Program of China under Grant No.2016YFB0800402.

References

[1] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population." in *Proc. FAST*, 2007, pp. 17–23.

[2] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you?" in *Proc. FAST*, 2007, pp. 1–16.

[3] J. Xiao, Z. Xiong, S. Wu, Y. Yi, H. Jin, and K. Hu, "Disk failure prediction in data centers via online learning," in *Proc. ICPP*, 2018, pp. 1–10.

[4] H. S. Gunawi, R. O. Suminto, R. Sears *et al.*, "Fail-slow at scale: Evidence of hardware performance faults in large production systems," in *Proc. FAST*, 2018, pp. 1–14.

[5] F. Mahdisoltani, I. Stefanovici, and B. Schroeder, "Proactive error prediction to improve storage system reliability," in *Proc. ATC*, 2017, pp. 391–402.

[6] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, "Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems," in *Proc. SIGMETRICS*, 2008, pp. 241–252.

[7] 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 Trans. on Storage*, vol. 4, no. 1, pp. 1:1–1:42, 2008.

[8] A. Oprea and A. Juels, "A clean-slate look at disk scrubbing." in *Proc. FAST*, 2010, pp. 57–70.

[9] B. Schroeder, S. Damouras, and P. Gill, "Understanding latent sector errors and how to protect against them," *ACM Trans. on Storage*, vol. 6, pp. 9:1–9:23, 2010.

[10] G. Amvrosiadis, A. Oprea, and B. Schroeder, "Practical scrubbing: Getting to the bad sector at the right time," in *Proc. DSN*, 2012, pp. 1–12.

[11] M. Baker, M. Shah, D. S. Rosenthal, M. Roussopoulos, P. Maniatis, T. J. Giuli, and P. Bungale, "A fresh look at the reliability of long-term digital storage," in *Proc. EuroSys*, 2006, pp. 221–234.

[12] J. Liu, K. Zhou, Z. Wang, L. Pang, and D. Feng, "Modeling the impact of disk scrubbing on storage system." *Journal of Computers*, vol. 5, no. 11, pp. 1629–1637, 2010.

[13] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," in *Proc. SIGMETRICS*, 2007, pp. 289–300.

[14] L. N. Bairavasundaram, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Characteristics, impact, and tolerance of partial disk failures," Ph.D. dissertation, University of Wisconsin–Madison, USA, 2008.

[15] J. Li, R. J. Stones, G. Wang, Z. Li, X. Liu, and K. Xiao, "Being accurate is not enough: New metrics for disk failure prediction," in *Proc. SRDS*, 2016, pp. 71–80.

[16] BACKBLAZE, "The backblaze hard drive data and stats." https://www.backblaze.com/b2/hard-drive-test-data.html, 2018, [Online; accessed 1-february-2018].

[17] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[18] J. Li, X. Ji, Y. Jia, B. Zhu, G. n. Wang, Z. Li, and X. Liu, "Hard drive failure prediction using classification and regression tre es," in *Proc. DSN*, 2014, pp. 383–394.

[19] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.