PAPER

Effects of Data Scrubbing on Reliability in Storage Systems

Junkil RYU[†a)], ***Nonmember and*** **Chanik PARK**[†b)], ***Member***

SUMMARY Silent data corruptions, which are induced by latent sector errors, phantom writes, DMA parity errors and so on, can be detected by explicitly issuing a read command to a disk controller and comparing the corresponding data with their checksums. Because some of the data stored in a storage system may not be accessed for a long time, there is a high chance of silent data corruption occurring undetected, resulting in data loss. Therefore, periodic checking of the entire data in a storage system, known as data scrubbing, is essential to detect such silent data corruptions in time. The errors detected by data scrubbing will be recovered by the replica or the redundant information maintained to protect against permanent data loss. The longer the period between data scrubbings, the higher the probability of a permanent data loss. This paper proposes a Markov failure and repair model to conservatively analyze the effect of data scrubbing on the reliability of a storage system. We show the relationship between the period of a data scrubbing operation and the number of data replicas to manage the reliability of a storage system by using the proposed model.

key words: *data reliability, data scrubbing, storage system*

1. Introduction

Today, hard disk drives are mainly used in enterprise computing and personal computing environments. Hence, the technical nature and trends in hard disk drives must be understood to build highly reliable storage systems. In previous research [9]–[11], complete disk drive failures were the main focus when analyzing the reliability of storage systems. Thus, the reliability model of storage systems has been based on the assumption that the data is mainly lost due to complete disk drive failures. However, recent studies [4], [5] show that factors other than complete disk drive failures influence data reliability.

Most of the current storage systems consider only complete disk drive failures when building highly reliable storage systems. However, the data reliability is also affected by silent data corruptions, which are induced by latent sector errors, phantom writes, misdirected reads and writes, DMA parity errors (which are the errors occurred when transferring data via DMA.) and so on [21], [22]. The redundant data plans based on complete disk drive failures are not adequate because silent data corruptions are not instantly reported to the storage system, so the repair process is delayed. A storage system does not detect the occurrence of the silent data corruptions until the corresponding sectors are accessed and the related data are verified explicitly, which is a data scrubbing operation. Latent sector errors, which are detected by explicitly issuing a read command to a disk controller, are the cause of most silent data corruptions in disk-based storage systems. The impact of latent sector errors on the reliability of storage systems has recently been recognized. A single latent sector error can lead to data loss during RAID group reconstruction after a disk drive failure [8].

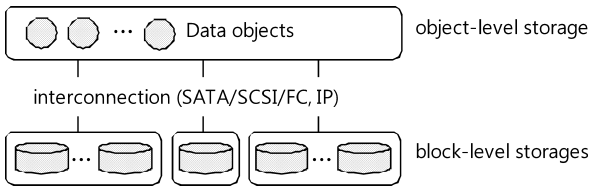


Fig. 1 Generic storage system architecture.

Figure 1 shows a generic storage system architecture, where the block-level storage system is a collection of hard disk drives and handles blocks (or sectors), and the object-level storage system is a logical storage system such as Google FS [19], ZFS [21], and Lustre [20], and handles data objects like data files. The data scrubbing operation can be performed in an object-level storage system as well as in a block-level storage system. The data scrubbing operation in this paper is assumed to be performed in an object-level storage system, which directly manages the data replication mechanism and its underlying block-level storage system is just a bunch of disk drives. Data scrubbing operation is composed of a data read operation fetching data parts from block-level storages and a data verification operation comparing the read data object with its checksum computed and stored previously. Silent data corruptions in this paper are defined as the errors which are not reported to an object-level storage system until the related sectors are accessed and the corresponding data object is verified with its checksum like ZFS [21]. Silent data corruptions are detected by data scrubbing operations in an object-level storage system and their recovery is done as a unit of a data replica. In this paper, we focus on latent sector errors, which is the main source of silent data corruptions. Latent sector errors are found out by a disk drive controller when the related sectors are accessed and they are reported to an object-level storage system.

Figure 2 shows a traditional Markov model of a stor-

Manuscript received December 1, 2008.
Manuscript revised April 6, 2009.

[†]The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk 790–784, Republic of Korea.
a) E-mail: lancer@postech.ac.kr
b) E-mail: cipark@postech.ac.kr

DOI: 10.1587/transinf.E92.D.1639

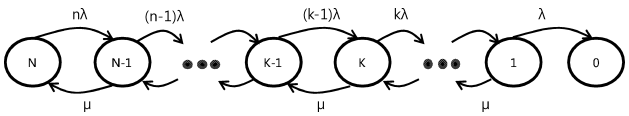


Fig. 2 Traditional Markov model of a storage system with n replications: λ is the failure rate of a data replica, μ is the repair rate of a data replica.

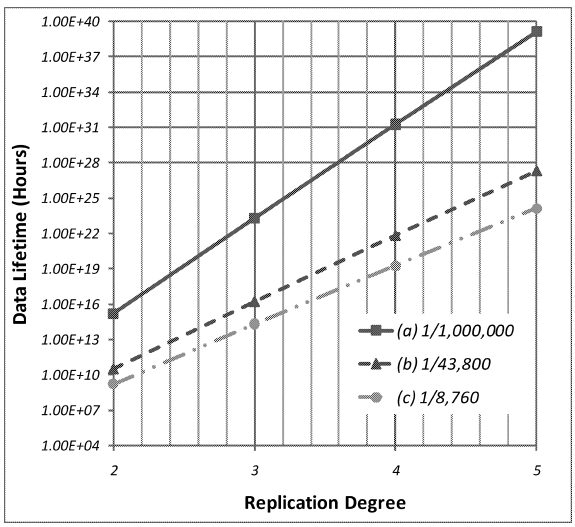


Fig. 3 Data lifetimes in each failure rate (λ): (a) 1/1,000,000, (b) 1/43,800, (c) 1/8,760.

age system with n replicas. It is assumed that each failure of a data replica is independent. This model assumes that the failure rate of data replicas is proportional to the number of currently available data replicas ($(n - k) \times \lambda$) and each repair rate has the same value (μ). This means that multiple data replicas can fail almost simultaneously but failed data replicas are repaired one by one. Thus, the model evaluates the data lifetime conservatively. Figrure 3 shows the data reliability (expected data lifetime, which is the mean time to data loss) according to the failure rate of a data replica. The repair rate is 36 objects per hour, where an object is 100 MB. This model and the equations for the expected data lifetimes are made by using the model and the equations in [6]. Each failure rate means that *1,000,000* is the MTTF (Mean Time To Failure) provided by a disk drive manual: *43,800* is the disk drive warranty time provided by a disk drive maker; *8,760* is the non-recoverable read error invoking time in the assumption that about 10 TB data will be read a year. The lifetime of line (a) in the Fig. 3 is much larger than line (b) and (c) because the failure rate of the former is much smaller and the recovery rate is the same. In the traditional Markov model, a data recovery process begins as soon as a data replica failure occurs. This scenario is possible when the data replica failure is caused by a complete disk failure, which is reported almost immediately. In the case of line (c), the loss of a data replica will be not reported until the corresponding disk sectors are checked. However, since the model assumes data failures are reported immediately, it exaggerates the reliability in the case of silent data corruptions.

Data scrubbing in storage systems is needed to recover data from redundant data when silent data corruptions occur due to component faults, media faults, and software faults. Disk based storage systems must have more frequent chances to check the data in storage systems than offline backup storage systems because the disk drives are almost always running and the chance of error in disk based storage systems is higher. How often must the data scrubbing operation be performed to ensure the reliability of a data storage system? Frequent data scrubbing operations will affect the storage service in terms of user response time and throughput. Infrequent data scrubbing operations will lead to permanent data loss.

This paper proposes a simple Markov model to determine the number of data replicas and the period of data scrubbing suitable for the required reliability of a disk-based storage system. The remainder of the paper is set out as follows. Section 2 explains the technical nature of hard disk drives and the types of silent data corruptions causes and related work. Section 3 proposes a Markov model of storage systems to show the effects of a data scrubbing operation on the reliability. Section 4 shows the feasibility of data scrubbing while not interrupting user data services with real I/O traces. Section 5 concludes the paper.

2. Background

2.1 Hard Disk Drives

Disk drives are the main data storage for both industrial and personal usage. While a growing number of disk drives are finding their way into mobile and consumer appliances (e.g., video recorders and personal electronics), disk drives for the computing industry are segmented into enterprise and personal applications. Also arising is a new segment called "nearline enterprise" that combines some of the characteristics of the classic personal and enterprise markets. Administrators consider price, capacity, interface type (e.g., ATA, SATA, SCSI, or Fibre Channel), performance (e.g., I/Os per second, response time, and sustained transfer rate), and reliability (e.g., MTBF, unrecoverable error rate) to satisfy application demands and user satisfaction. Each attribute of a disk drive carries varying weight depending on the workload which the applications generate. Figure 4 shows the characteristics of disk drives according to class. While the capacity of personal and nearline class disk drives reach 1 TB, the enterprise class disk drives remain at low-capacity, at most 300 GB. A disk drive needs more disks to have higher capacity. More disks in a disk drive reduce the space between disks and increase the possibility of contact with disk heads. Also, higher speeds raise the temperature in a disk drive and increase data loss. Enterprise class disk drives have higher speeds than personal and nearline class disk drives. Hence, enterprise disk drives cannot have as high capacity as personal and nearline disk drives because of the data reliability requirements. While SATA disk drives can be made as ef-

	Enterprise disk drives	Nearline enterprise disk drives	Personal disk drives
Interface Type	SCSI, FC, SAS	SATA	ATA, SATA
Rotational Speed (rpm)	10,000 ~ 15000	7,200	4,800 ~ 7,200
Capacity	~ 300GB	~ 1 TB	~ 1TB
AFR (%)	~ 0.62	0.34	0.34 ~
Non-recoverable errors per bits read	1 sector per 10^{15} ~ 10^{16}	1 sector per 10^{14} ~ 10^{15}	1 sector per 10^{14}
Sustained Transfer Rate	~ 125 MBPS	~ 78 MBPS	~ 78 MBPS
Average Seek Time	3.5 ~ 4 ms	8.5 ~ 10 ms	8.5 ~ 10 ms

Fig. 4 Disk drive characteristics.

fective and reliable as SCSI/SAS/FC disk drives with similar capacities, their costs would be comparable. Reliability and performance come not from the interface but from drive design and testing. This tradeoff between cost and reliability is seen in many other electronics. The electronic parts in enterprise disk drives have a significantly higher investment in reliability design and testing than those in personal and nearline disk drives.

When building large-scale storage systems, SATA disk drives can have a reliability advantage because enterprise disk drives such as SCSI/SAS/FC have lower-capacity (174 GB is common today and the maximum capacity is 300 GB). 1 TB capacity in SATA disk drives is becoming common. Hence this capacity gap allows SATA disk drives' failure rate to be up to six times the SCSI failure rate. SCSI failure rates are specified and tested for the worst allowable conditions. The most significant difference in the reliability specification of personal and enterprise class disk drives is the expected power-on hours for each drive type. The AFR (Annual Failure Rate) calculation for personal and nearline disk drives assumes 8 hours/day for 300 days/year while that for enterprise disk drives assumes 24 hours/day for 365 days/year [13]. The lower the annual failure rate, the longer a disk drive is expected to run. Hence the MTBF (mean time between failures) calculation for a personal disk drive is 705,800 ($300 \times 8/0.34\%$) hours while the MTBF calculation for an enterprise disk drive is 1,412,903 ($365 \times 24/0.62\%$) hours. According to field data, annual disk replacement rates typically exceed 1%, with 2-4% common and up to 13% observed on some systems [11]. The gap between disk drive specification and field data varies depending on temperature, humidity, workloads, etc. To build storage systems satisfying the demanded data reliability, administrators must select a disk drive considering these differences between the specification and field data.

With technical improvements, hard disk drives are increasing their capacity and media access rates. However, contrary to this trend, the I/O bus bandwidth such as SATA and SCSI, is increasing at a low rate. Also the chances of having latent sector errors in hard disk drives are increasing linearly with disk drive capacity. These factors increase the possibility of data loss induced by latent sector errors. The studies in [12] and [14] related with file I/O traces show that a relatively small percentage of files stored in a storage system are active. In [14], about 5% of the total disk storage was accessed over a 24 hour period. The disk storage size of *cello*, which had the highest capacity, was 10.4 GB. Currently, the size of hard disk drives reaches up to 1 TB and the active portion of the data stored in storage systems is becoming relatively smaller as the capacity of storage systems increases. The number of latent sector errors is increasing as the active portion of data stored in storage systems is decreasing. Therefore, the reliability of high capacity storage systems is becoming worse.

Non-recoverable errors per bits read of 1 sector per 10^{14} on nearline enterprise disk drives means that a read failure (or latent sector error) occurs about every 10 TB. This means that the data loss possibility in the course of rebuilding a storage system comprised of 5×500 GB disk drives in RAID 5 when a complete disk drive failure occurs is about 20% [1]. However this value is calculated assuming that failures on a disk drive are detected as soon as they occur. In most cases, non-recoverable read errors on a disk drive are not detected as quickly as a complete disk drive failure. If a sector on a disk drive was written onto and has not been read from since, its error occurrence can not be known; this can induce permanent data loss. Considering that most nearline enterprise disk drives have non-recoverable error rate of 10^{-14} and their capacity is about 1 TB, if 10 TB of data is read per year (a reasonable assumption), a data loss incident can happen every year. Therefore a non-recoverable read error rate as well as a disk drive failure rate (AFR) should be considered for reliable storage systems.

2.2 Data Loss Cases

The following cases show that silent data corruptions can occur at different system layers, i.e., the file system layer, device driver layer, storage device layer, etc.

Component faults: A storage system is composed of many electronic components such as controller, memory, and disk drives. These electronic components can fail due to power surges, temporary power loss, environmental factors (e.g., temperature, humidity, dust), and human mistakes. These component failures affect the data reliability of storage systems by inducing partial or complete data failures. To prepare for these component faults, the system designer includes extra electronic parts in a storage system. An example is a RAID system, which includes redundant disk drives to prepare for complete disk failures.

Disk media faults: The causes of latent sector errors include: disk media imperfections, read/write head contacts with disk media, and high-fly inducing incorrect write. Disk media imperfections are excluded mainly by using the extra sectors in a disk in factory tests. Current disk drives use such things as stronger arm assembly, overcoats, and vibration sensors to avoid read/write head contacts with disk media. However, for capacity and performance, more platters and faster revolution are needed, which increase the possibility of the head contacts. These head contacts affect large adjacent sectors on the same disk and track [16]. Hence, an

unrecoverable read error induced by a head contact can include many sectors.

We gathered Self-Monitoring Analysis and Reporting Technology (SMART [25]) information from the disk drives running in public-use computers installed in our campus library. Sixty-three disk drives were analyzed and 11 disk drives had unrecoverable read errors. The manufacturers of the disk drives with these errors included Seagate, Western Digital, Maxtor, and Samsung. According to the gathered field data, if an unrecoverable read error is found in a disk drive, then the number of affected adjacent sectors ranges from 1 to 64, with 8 sectors being the most common. SMART cannot provide the total unrecoverable read errors and the error rate for a disk drive, due to SMART record limitations and localized data access patterns. As the storage capacity and performance increase, the possibility and data damage of a head contact increase. Also, these errors can cause permanent data losses because the unrecoverable read errors are not instantly reported.

Software faults: The software stack of storage systems, which includes a file system, a device driver, and firmware within a device driver, is not perfect. The imperfections of the software stack are revealed by the faults happening in the entire I/O path such as phantom writes, misdirected reads and writes, DMA parity errors, driver bugs, and accidental overwrites. The field data and the analysis of these faults in the software stack are explained in [18], and a parity-based technique preventing the data losses induced by these faults is introduced in [17].

2.3 Related Work

Shwarz et al. [2] recognized the data loss possibility due to the latent sector errors in the disk drives forming a MAID (Massive Array of mainly Idle Disks) system and proposed a disk scrubbing technique. This is the same as the data scrubbing operation done in block-level storage in Fig. 1. In this research, efficient disk scrubbing algorithms such as random, deterministic, and opportunistic scrubbing to reduce the power consumption of disk drives within a MAID were introduced and the reliability of two-way mirroring system was evaluated with a simple Markov model. Kotla et al. [3] recognized that permanent data losses occur because the system did not report the loss of replicas to the user and did not actively repair them. They proposed a data checking and recovering mechanism to prevent permanent data loss due to those reasons. This research implemented the mechanism but did not explain in detail the relationship between data reliability and the data scrubbing operation. Hence, we propose a simple Markov model applied to general storage systems to analyze the relation between data reliability and data scrubbing. Baker et al. [5] considered latent faults and temporal correlation of faults in their reliability analysis model on a storage system with two replicas - mirrored data. A number of strategies for improving reliability of a storage system have been suggested in their paper using their model, but without further detailed analysis. Kari et al. [23], [24] introduced an adaptive algorithm utilizing the idle time of the disk for scanning commonly used disks to detect latent sector faults.

3. Data Scrubbing

3.1 Our Proposed Markov Model

The storage system in this paper is an object-level storage and manages replicas without block-level storage help, and disk drives are connected by IP/FC network as well as the bus such as SATA and SCSI, and each replica is stored in a different disk drive. Each replica failure is assumed to be independent in this model, where latent sector errors in different disk drives are not correlated. According to [4], the latent sector errors develop aggressively in the initial 50 days and then stabilized with enterprise disk drives. When considering the latent sector errors in the enterprise disk drives, latent sector errors in different disk drives can be reasonably assumed to follows the exponential distribution. Hence, it can reasonably be assumed that replica failures due to latent sector errors follow the exponential distribution. Without loss of generality, we consider a simple method of data replication to generate the redundant information. Since it is possible to detect the occurrence of silent data corruptions by data scrubbing, we propose a simple Markov model to consider the rate of a data scrubbing operation for reliability analysis of a storage system. Figure 5 shows how silent data corruptions are developed and detected by a data scrubbing operation. The repair process will follow to recover the detected silent data corruptions. Without a data scrubbing operation, the number of failed replicas induced by silent data corruptions will grow indefinitely before they are detected because the storage system does not recognize that there are failed data replicas. A data scrubbing operation in time reports silent data corruptions to the storage system, triggers a repair process of the failed data replicas induced by silent data corruptions, and as a result, prevents permanent data losses.

Silent data corruptions are caused by latent sector errors, DMA parity errors, misdirected reads/writes, and phantom writes. However, not all of the silent data corruptions can be detected and recovered at block level. For ex-

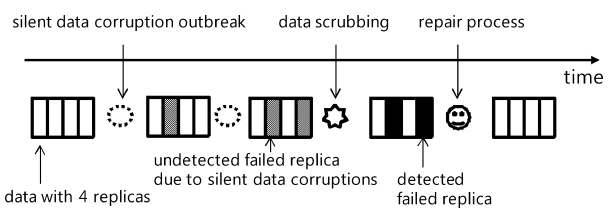


Fig. 5 Data scrubbing and repair process: Silent data corruptions remain undetected until data scrubbing is executed over the corresponding sectors. Shaded boxes represent the occurrence of silent data corruptions, but undetected. After data scrubbing, shaded boxes are turned black, meaning they are detected. Then, the repair process begins to execute depending on management policy.

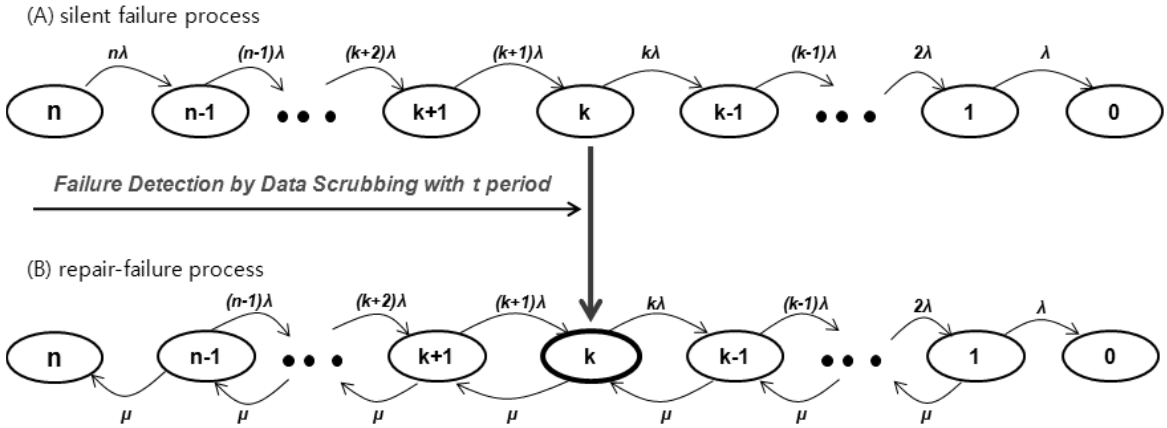


Fig. 6 Our proposed Markov model with silent data corruptions: data scrubbing operation occurs every t hours, n is the number of data replicas, μ is the repair rate of failed replicas, and λ is the failure rate due to silent data corruptions. (A) The silent failure process determines the number of failed data replicas due to silent data corruptions. The figure shows that n-k silent data corruptions are developed during t. (B) The repair-failure process is a traditional Markov model, where a repair operation is performed as soon as a replica fails because the related replica failures due to silent data corruptions are detected during the recovery process.

ample, the silent data corruption caused by latent sector errors can be detected and recovered at block level whereas the silent data corruption caused by DMA errors, misdirected reads/writes, and phantom writes cannot be detected at block level due to insufficient information. In order to detect the silent data corruption caused by DMA errors, misdirected reads/writes, and phantom writes, we need to maintain some additional information like checksum at a higher level (e.g. object-level) than block-level. For simplicity, we assume an object-level storage system where all of the silent data corruptions can be handled at object-level. There may be some overhead to recover a latent sector error because the entire replica of an object should be moved. However, this overhead can be controlled by configuring the size of an object to any size. Because the reliability analysis is a main focus of the paper, we will look at how the object size affects the reliability of a storage system in Sect. 3.3.3.

Our proposed Markov model consists of two separate processes called the silent failure process (A) and the repair-failure process (B) as shown in Fig. 6. The complete disk drive failure rate (about 1×10^{-6}) is much lower than the latent sector error rate (about 1×10^{-4}). That is, the reliability of a storage system will be determined dominantly by the latent sector error rate. The silent failure process determines the number of failed data replicas induced by silent data corruptions when they are detected by a data scrubbing operation. And the repair-failure process is a traditional Markov model as shown in Fig. 2 [9]. The role of the silent failure process is to designate the initial state in the repair-failure process after waiting for time t, an execution period of data scrubbing operation. Hence, the initial state in the repair-failure process may not be the state n, whereas the state n is always the initial state in a traditional Markov model. If $t = 0$, i.e. silent data corruptions can be detected immediately without any delay, our proposed model degenerates to the traditional Markov model shown in Fig. 2. Note that the state n in the repair-failure process does not have a failure process but state n in the silent failure process has a failure process. Repairing all failed replicas means that repair operations induced by a data scrubbing execution end and then new silent data corruptions can occur. In Fig. 6 (A), λ is the rate of a latent sector error and the transition from state k to state k-1 is assumed to be made by the rate of $k \times \lambda$ where k is the number of replica currently available. Note that the chance is extremely low to witness latent sector errors developed for the sectors among the replicas for an object. However, because the reliability analysis should be done conservatively and we need to consider any chance of non-zero probability in the analysis, we can assume that the transition from state k to state k-1 is made at the rate of $k \times \lambda$. The failure rates, λ, are in proportion to the number of currently available data replicas and the repair rate is the same as μ in Fig. 6 (B). These parameters are configured to measure the reliability of a storage system conservatively.

3.2 Analytical Model

The equations for the expected data lifetime in disk drive based storage systems are derived as follows. In the derivation of the equations, we use the concept of an *epoch*. The start of an epoch is when the latent data errors develop and the end of an epoch is either when the system returns to state n or when the system goes to state 0. If the system returns to state n, it means that the next epoch starts. N_k is the expected number of the epochs when the repair process of the repair-failure process starts in state k until the system goes to state 0. T_k is the expected duration of each epoch when the recovery process of the system starts in state k. P_k is the probability that the system is in state k when the data scrubbing with period t is applied. Let Q_k be the probability that

the system reaches state 0 before state n starting in state k.

$$P_k = \frac{n!}{(n-k)!k!} \times \left(e^{-\lambda t}\right)^k \left(1 - e^{-\lambda t}\right)^{n-k} \quad (1)$$

$$Q_k = \frac{k\lambda}{k\lambda + \mu} Q_{k-1} + \frac{\mu}{k\lambda + \mu} Q_{k+1}$$

$$\left(Q_0 = 1,\ Q_n = 0,\ p_k = \frac{k\lambda}{k\lambda + \mu},\ q_k = \frac{\mu}{k\lambda + \mu}\right)$$

The following equations are the derivation of Q_k and N_k.

$$\begin{aligned}(p_k + q_k)Q_k &= p_k Q_{k-1} + q_k Q_{k+1} \\ Q_{k-1} - Q_k &= \frac{q_k}{p_k}(Q_k - Q_{k+1}) \\ &= \left(\frac{\mu}{k\lambda}\right)(Q_k - Q_{k+1}) \\ Q_{n-1} - Q_n &= Q^*\end{aligned}$$

Applying the above equation recursively,

$$Q_{n-k-1} - Q_{n-k} = \frac{1}{\mathbf{P}(n-1,k)} \times \left(\frac{\mu}{\lambda}\right)^k \times Q^*$$

Here, P(n, k) means a permutation calculation: $\frac{n!}{(n-k)!}$.

$$\begin{aligned}Q_0 - Q_n &= Q^* \sum_{j=0}^{n-1} \frac{1}{\mathbf{P}(n-1,j)} \times \left(\frac{\mu}{\lambda}\right)^j \\ Q^* &= \left(\sum_{j=0}^{n-1} \frac{1}{\mathbf{P}(n-1,j)} \times \left(\frac{\mu}{\lambda}\right)^j\right)^{-1}\end{aligned}$$

Obtaining Q_k by applying Q^* to the above recursive equation,

$$Q_k = Q^* \sum_{j=0}^{n-k-1} \frac{1}{\mathbf{P}(n-1,j)} \times \left(\frac{\mu}{\lambda}\right)^j \quad (2)$$

So,

$$N_k = \frac{1}{P_k \times Q_k} \quad (3)$$

The following equations are the derivation of T_k.

$$\begin{aligned}T_k &= p_k T_{k-1} + q_k T_{k+1} + \frac{1}{k\lambda + \mu} \\ T_{n-1} - T_n &= T^*\end{aligned}$$

Applying the above equation recursively,

$$\begin{aligned}T_{n-k-1} - T_{n-k} &= \frac{1}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)T^* \\ &\quad -\frac{1}{n\lambda}\sum_{j=1}^{k}\frac{\mathbf{P}(n,j)}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)^{k-j}\end{aligned}$$

$$\begin{aligned}T_0 - T_n &= \sum_{k=0}^{n-1}(T_{n-k-1} - T_{n-k}) \\ &= \sum_{k=0}^{n-1}\left(\frac{1}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)^k T^*\right) \\ &\quad -\frac{1}{n\lambda}\sum_{k=0}^{n-1}\sum_{j=1}^{k}\frac{\mathbf{P}(n,j)}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)^{k-j}\end{aligned} \quad (4)$$

$$T^* = Q^*\left(\frac{1}{n\lambda}\sum_{k=0}^{n-1}\sum_{j=1}^{k}\frac{\mathbf{P}(n,j)}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)^{k-j}\right)$$

Obtaining T_k by applying T^* to the above recursive equation,

$$\begin{aligned}T_k &= \sum_{j=0}^{n-k-1}\left(\frac{1}{\mathbf{P}(n-1,j)}\left(\frac{\mu}{\lambda}\right)^j T^*\right) \\ &\quad -\frac{1}{n\lambda}\sum_{j=0}^{n-k-1}\sum_{m=1}^{j}\frac{\mathbf{P}(n,m)}{\mathbf{P}(n-1,k)}\left(\frac{\mu}{\lambda}\right)^{j-m}\end{aligned} \quad (5)$$

The expected data lifetime is $\sum_{k=0}^{n-1} T_k \times N_k + T$, where T is the sum of the times remaining in state n.

3.3 Reliability Analysis

The following analysis uses $\frac{1}{8{,}760}$ as the failure rate (8,760 is 24 hours × 365 days) and 36 objects per hour as the repair rate, where the size of an object is 100 MB. Figure 7 shows the probability of a system being in state k after waiting the specified scrubbing interval; here, k is the number of valid replicas remaining when the replication degree is 3 and the plot shows the change of each state as a scrubbing interval increases. As the scrubbing interval increases, the possibility of state 3 decreases, and that of state 2 increases up to about 3,000 hours and then decreases, that of state 1 increases up to about 10,000 hours and then decreases. Figure 8 shows that the expected replicas decrease linearly as the scrubbing interval increases, where the total number of the replicas is 3. A system which does not

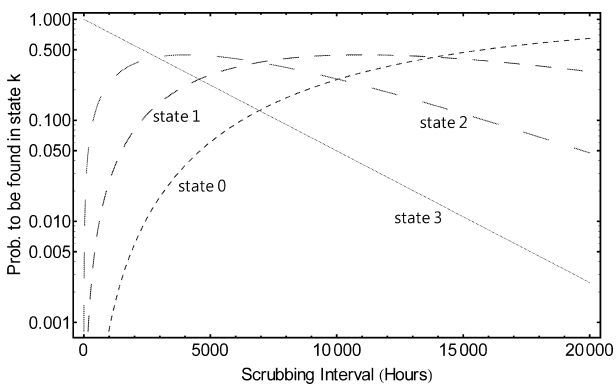


Fig. 7 Probability to be found in state k after scrubbing interval t when the replication degree of a storage system is 3.

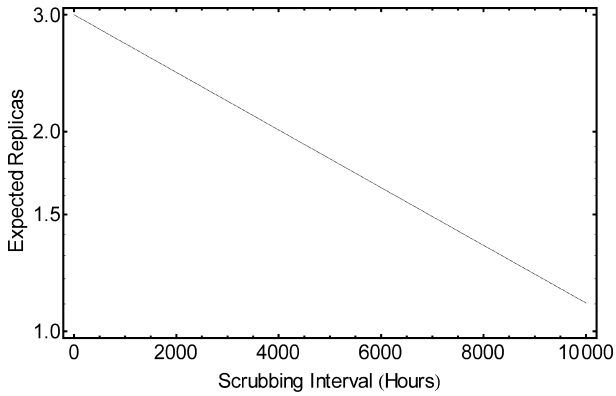


Fig. 8 The expected number of the live replicas after scrubbing interval t when the replication degree of a storage system is 3.

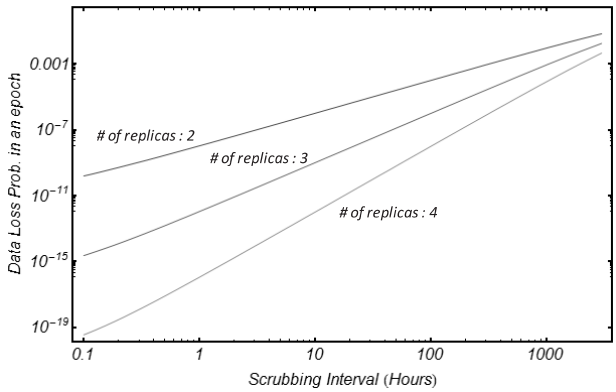


Fig. 9 Data loss probability of an epoch at various replication degrees.

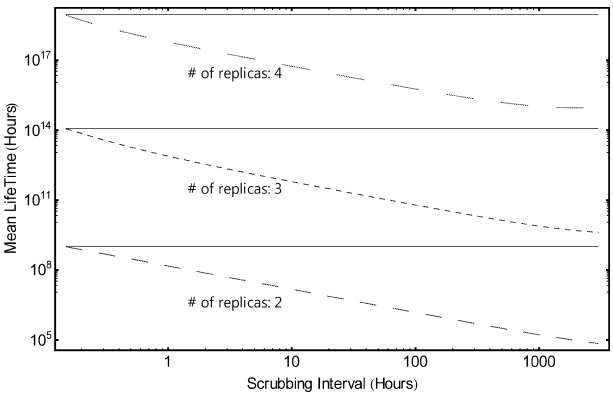


Fig. 10 Data lifetimes at various replication degrees, where each solid line represents data lifetime when data scrubbing time is 0.

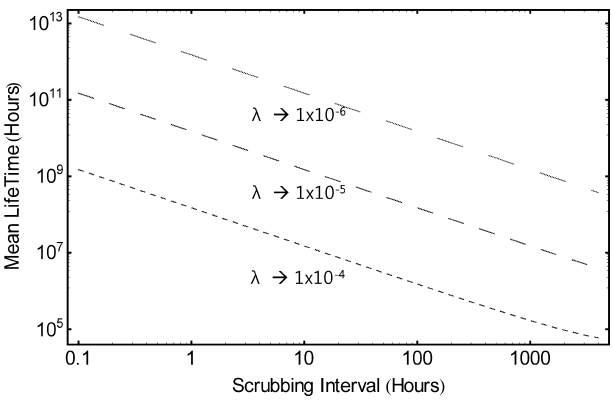


Fig. 11 Data lifetimes with 2 replicas at failure rates of 1×10^{-4}, 1×10^{-5}, and 1×10^{-6}.

identify and remove latent errors can not properly utilize the data reliability of disk drives, which have MTBFs of 700,000 to 1,400,000 hours. If a data scrubbing operation is applied in a storage system, how often should it be done? A data scrubbing operation consumes system resources such as disk bandwidth, system bus and memory and can occasionally interrupt the user data service. Hence, the frequency of a data scrubbing operation should be as small as possible while maintaining the data reliability. Figure 9 shows that the data loss probability in an epoch when the number of the replicas is 2, 3, and 4 increases as the scrubbing interval increases. The larger the number of data replicas, the smaller the data loss probability in an epoch. To improve the data reliability, decreasing the scrubbing interval is effective. Figure 10 is the expected data lifetime when a data scrubbing operation is applied, where the solid lines are the expected data lifetime when the silent data corruptions are detected as soon as they develop. The data lifetime decreases rapidly as the data scrubbing interval increases when the scrubbing interval is small but the data lifetime decreases very slowly as the data scrubbing interval increases when the scrubbing interval is large. The difference of the expected data lifetime is very large when considering or not considering the silent data corruptions. When three replicas are used, the model that does not consider the silent data corruptions predicts the expected data lifetime to be about 1×10^{14} hours, but the model including the silent data corruptions predicts the expected data lifetime to be about 6×10^{10} hours when the scrubbing interval is 100 hours.

This probability model helps measure the data reliability because all errors are not revealed externally as soon as they develop. As the scrubbing interval increases, the reduction in the expected data lifetime becomes smaller. Changing the scrubbing interval in the range of high values does not largely affect the data reliability. Hence, controlling the data reliability with a scrubbing interval is effective only when the interval value is small. At larger numbers of replicas, using more replicas results in a smaller decrease in expected lifetime as the scrubbing interval increases, even at smaller intervals. The slope with 4 replicas is gentle at 1000 hours but that with two replicas is not. Setting the complete disk drive failure rate to a different value from the silent data corruption rate only shifts each line up or down in the plot but does not change the shape of each line.

Figure 11 shows the expected data lifetimes at failure rates of $1\times10^{-4}, 1\times10^{-5}$, and 1×10^{-6}. The expected data lifetime is larger when the failure rate is smaller, and the slope of each failure rate is almost the same. To control the reliability and the performance of storage systems, we

can use parameters such as scrubbing interval, number of replicas, and the repair rate (μ). Note that the failure rate (λ) is not included because it is a property of the disk drives in the system.

3.3.1 Scrubbing Interval vs. The Number of Replicas

For data reliability, a data scrubbing operation must block user data service or be performed with the requests of user data services if the system does not have sufficient idle time. The average system overhead induced by a data scrubbing operation is in proportion to

$$\frac{size\ of\ target\ objects\ (S_{Obj})}{scrubbing\ bandwidth\ (B_{Sc})} \times \frac{1}{T_{Sc}},$$

where scrubbing bandwidth is allocated for the scrubbing and the target objects' scrubbing operations must be performed along with the user data services due to the lack of idle time, and T_{Sc} is the scrubbing interval. If the system does not have sufficient computing resources or its idle time is too short, increasing the scrubbing interval with additional replicas may be a good option. However, adding a replica may not achieve the expected data reliability because it increases the S_{Obj}. $\frac{S_{Obj}}{T_{Sc}}$ is expressed as λ_{Sc} and B_{Sc} is as μ_{Sc}. If $\lambda_{Sc} < \mu_{Sc}$, the system can process the demanded data scrubbing jobs. Otherwise, the system must increase T_{Sc} or B_{Sc} for the user data service and the data reliability. In most cases, increasing B_{Sc} is limited for user data services. Hence, the system must increase T_{Sc} for user data services and increase the number of replicas for the data reliability.

3.3.2 Scrubbing Interval vs. Failure Rate

Data failure rate can not be tuned by the system because it is a property of the storage system. Hence the interval of the data scrubbing must be configured short enough to make the existing storage system's reliability match the new storage systems built with more reliable parts such as recent SCSI/FC disk drives, and controllers.

3.3.3 Scrubbing Interval vs. Repair Rate

The repair rate is not directly related to the scrubbing interval in terms of data reliability. The repair rate does not participate in determining the initial state in the repair-failure process shown in Fig. 6. There is no repair process in the silent failure process either. However, Fig. 12 shows that the lifetime decreases rapidly as the scrubbing interval increases when the repair rate is small. A storage system with a larger repair rate is more reliable. Repair rate becomes more important as the number of failed data replicas to be repaired increases. Figure 13 shows the ratios between repair rates of 36 and 360, where we know that controlling repair rate is more effective when there are many replicas. When a storage system has 2 replicas, controlling its repair rate is not very effective.

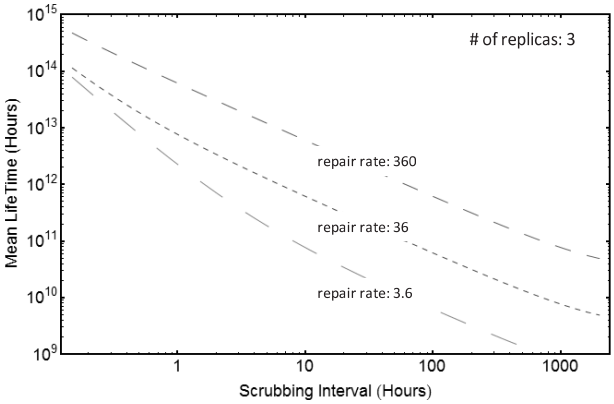


Fig. 12 Mean lifetimes according to repair rates of 3.6, 36, and 360.

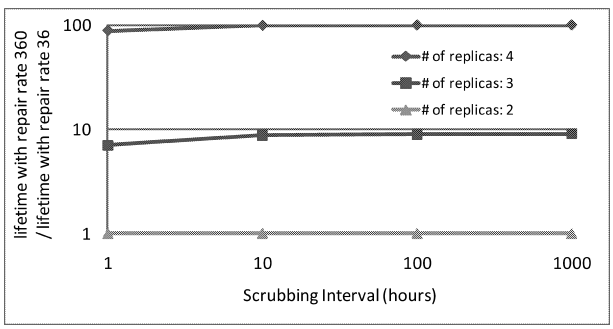


Fig. 13 Mean lifetime ratio between repair rates (36, 360) according to scrubbing intervals.

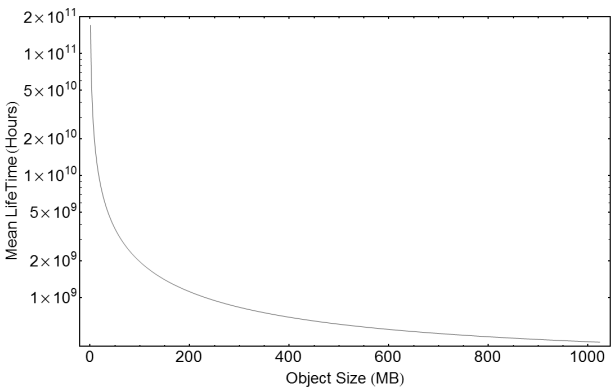


Fig. 14 Mean lifetime according to the object size, where repair bandwidth, scrubbing interval, and the number of replicas are 100 MB/sec, 100 hours, and 3 respectively.

The repair rate is dependent on the repair bandwidth and the object size. In order to increase the repair rate, we have to increase the repair bandwidth or decrease the object size. Note that increasing the repair bandwidth can interfere with normal data I/O services during the repair process. Figure 14 shows that the mean lifetime increases as the object size decreases. However, decreasing object size may increase the number of small I/O requests in a storage system, resulting in the degradation of the overall performance of the storage system.

4. Data Scrubbing Application

When building a reliable storage system, there are many options to be determined in advance such as replication degree, repair rate, and scrubbing interval. Figure 15 shows the expected data lifetime according to replication degree, repair rate, and scrubbing interval. A high replication degree means that the storage system must have more space for the redundant data. A large repair rate means that user data services can be blocked by the repair process when data failures occur. A small scrubbing interval means that data scrubbing operations can block user data services when its workload does not have sufficient idle time. A storage system with high reliability, availability and performance requires a large number of replicas and a large scrubbing interval, with a small repair rate. If we build a storage system with 3 replicas as reliable as a storage system with 4 replicas, scrubbing interval time 1000 hours and repair rate 3.6, we can set less than 100 hours for scrubbing interval, and more than 3600 for repair rate. However, these settings deteriorate user data services' quality such as a performance and the availability.

In Fig. 16, two I/O workloads (Financial1 and Financial2) were obtained from OLTP applications running at two large financial institutions, and one I/O workload (WebSearch2) was obtained from a popular search engine [15].

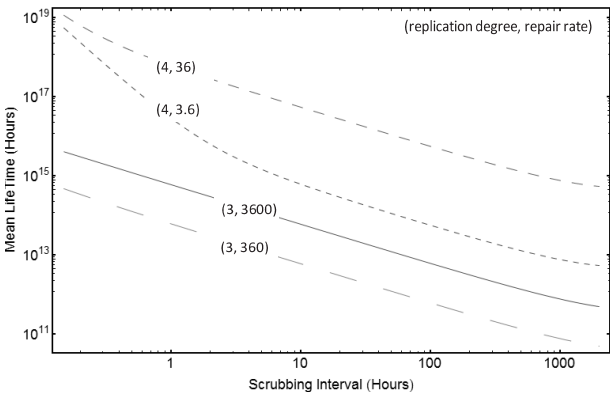


Fig. 15 Applying data scrubbing technique to storage systems.

ASU is an application specific storage unit. And each trace is obtained from the first ASU in each workload. A data scrubbing operation needs sufficient idle time in the workload so as not to interrupt and deteriorate user data services. The total idle time found in trace (a), (b) and (c) is about 4.3 hours, about 2.1 hours, and 0.6 hours during 12 hours of traces, respectively. Note that the idle time longer than 20 seconds is considered. The trace is collected during 12 hours, but we assume that the pattern of each trace is repeated afterwards. For furthur analysis, we assume the followings:

- The number and the size of objects are all the same in each trace, i.e. 48,000 and 100 MB, respectively.
- The number of replicas and the repair rate are 3 and 36 objects per hour.
- It takes 25 hours to complete the data scrubbing operations.

Then, lets consider a RAID 5 storage system consisting of five disks where each disk has 1 TB size and 1×10^{-6} in disk failure rate. If the average repair rate for disk failures is assumed to be 60 MB/sec, then we can get the mean data lifetime of 1×10^{10} hours. When we consider the silent data corruptions such as latent sector errors, the mean data lifetime will be reduced to about 1×10^5 because the latent sector error rate is much larger than the disk failure rate. However, in order to maintain the mean data lifetime unchanged as 1×10^{10} hours even with consideration of latent sector errors, we can determine from our model that the data scrubbing operation should be complete within 100 hours, that is, the data scrubbing interval should be 100 hours. Hence, we can see that the required mean data lifetime of 1×10^{10} can be supported in Trace (a) without interfering with normal user I/O services because the total idle time will be longer than 25 hours during the trace of 100 hours. However, in case of Trace (b) and (c), the storage system has to harvest some computation time for data scrubbing operation, that is, interrupting normal user I/O services, because of insufficient idle time for data scrubbing operations. However, if a system administrator does not want data scrubbing to interrupt normal user I/O services when the storage system does not have sufficient idle time, then the administrator may set

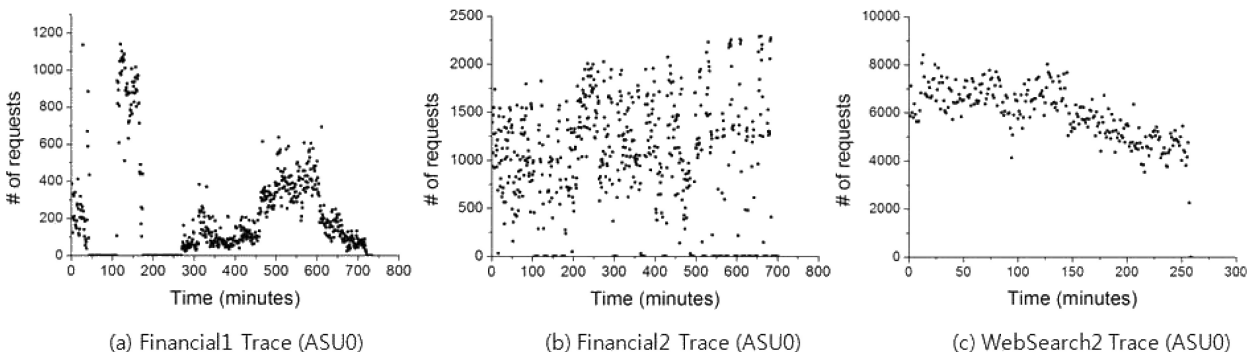


Fig. 16 The number of requests per minute according to time in Financial1, Financial2, and WebSearch3 [15].

the interval to be infinite, running the data scrubbing operations only when the storage system is idle. In that sense, our model compliments the traditional approach to invoke the data scrubbing operation.

5. Conclusion

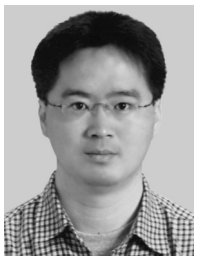
In this paper, we have addressed the importance of dealing with silent data corruptions due to phantom writes, misdirected read and writes, DMA parity errors, latent sector errors, and etc. And we have shown the necessity of data scrubbing to prevent permanent data loss. For data reliability, it is important that data scrubbing has an adequate interval under the given number of data replicas, repair rate of the storage system, and failure property of the storage system. The expected data lifetime increases as the data scrubbing interval decreases. We showed that user demanded data reliability can be satisfied by adjusting the configuration options such as the number of replicas, repair rate, and data scrubbing interval according to the limitations of system resources. This paper's contributions are as follows:

- We developed a simple Markov model that expresses the essential features of a storage system with the data scrubbing operation, applying a silent failure process into a traditional Markov model.
- We analyzed our proposed model, including the data loss possibility and the expected data lifetime according to the number of replicas, failure rate, repair rate, and scrubbing interval.
- We provided a method to satisfy the user demanded reliability in spite of the limitations of system resources.

In future work, we will devise a mechanism to periodically check the entire data in a storage system even if its capacity is very large by applying I/O access patterns of its workloads to a data scrubbing technique.

References

[1] G.F. Hughes and J.F. Murray, "Reilability and security of RAID storage systems and D2D archives using SATA disk drives," ACM Trans. Storage, vol.1, no.1, pp.95–107, Dec. 2004.
[2] T.J.E. Schwarz, Q. Xin, E.L. Miller, D.D.E. Long, A. Hospodor, and S. Ng, "Disk scrubbing in large archival storage systems," 12th IEEE/ACM Int'l. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Volendam, Netherlands, Oct. 2004.
[3] R. Kotla, L. Alvisi, and M. Dahlin, "SafeStore: A durable and practical storage system," 2007 USENIX Annual Technical Conference, June 2007.
[4] L.N. Bairavasundaram, G.R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," ACM SIGMETRICS the Int'l. Conference on Measurement and Modeling of Computer Systems, June 2007.
[5] M. Baker, M. Shah, D.S.H. Rosenthal, M. Roussopoulos, P. Maniatis, T. Giuli, and P. Bungale, "A fresh look at the reliability of long-term digital storage," EuroSys2006, April 2006.
[6] S. Rmabhadra and J. Pasquale, "Analysis of long-running replicated systems," INFOCOM 2006, 25th IEEE Int'l. Conference on Computer Communications, April 2006.
[7] Q. Lian, W. Chen, and Z. Zhang, "On the impact of replica placement to the reliability of distributed brick storage systems," 25th IEEE Int'l. Conference on Distributed Computing Systems, June 2005.
[8] J.L. Hafner, V. Deenadhayalan, K. Rao, and J.A. Tomlin, "Matrix methods for lost data reconstruction in erasure codes," USENIX Conference on File and Storage Technologies, Nov. 2005.
[9] W.A. Burkhard and J. Menon, "Disk array storage system reliability," 23rd International Symposium on Fault-Tolerant Computing, 1993.
[10] E. Pinheiro, W.D. Weber, and L.A. Barroso, "Failure trends in a large disk drive population," USENIX Conference on File and Storage Technologies, Feb. 2007.
[11] B. Schroeder and G.A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?," USENIX Conference on File and Storage Technologies, Feb. 2007.
[12] K.K. Ramakrishnan, P. Biswas, and R. Karedla, "Analysis of file I/O traces in comercial computing environments," Performance Evaluation Review, vol.20, no.1, pp.78–90, June 1992.
[13] D. Anderson, J. Dykes, and E. Riedel, "More than an interface - SCSI vs. ATA," USENIX Conference on File and Storage Technologies, March 2003.
[14] C. Ruemmler and J. Wilkes, "A trace-driven analysis of disk working set sizes," Technical Report HPL-OSR-93-23, Hewlett-Packard Laboratories, April 1993.
[15] Two I/O traces from OLTP applications running at two large financial institutions, http://www.storageperformance.org, 2008.
[16] H. Huang and K.G. Shin, "Partial disk failures: Using software to analyze physical damage," 24th IEEE Conference on Mass Storage Systems and Technologies, Sept. 2007.
[17] A. Krioukov, L.N. Bairavasundaram, G.R. Goodson, K. Srinivasan, R. Thelen, A.C. Arpaci-Dusseau, and R.H. Arparci-Dusseau, "Parity lost and parity regained," USENIX Conference on File and Storage Technologies, Feb. 2008.
[18] L.N. Bairavasundaram, G.R. Goodson, B. Schroeder, A.C. Arpaci-Dusseau, and R.H. Arpaci-Dusseau, "An analysis of data corruption in the storage stack," USENIX Conference on File and Storage Technologies, Feb. 2008.
[19] S. Ghaemawat, H. Gobioff, and S. Leung, "The Google file system," 19th ACM Symposium on Operating Systems Principles, Oct. 2003.
[20] Luster file system, http://wiki.lustre.org
[21] ZFS, http://opensolaris.org/os/community/zfs
[22] V. Prabhakaran, L.N. Bairavasundaram, N. Agrawal, H.S. Gunawi, A.C. Arpaci-Dusseau, and R.H. Arpaci-Dusseau, "IRON file systems," Twentieth ACM Symposium on Operating Systems Principles, Oct. 2005.
[23] H.H. Kari, H. Saikkonen, and F. Lombardi, "Detection of defective media in disks," IEEE Int'l. Workshop on Defect and Fault Tolerance in VLSI systems, Oct. 1993.
[24] H.H. Kari, Latent Sector Faults and Reliability of Disk Arrays, Ph.D. Thesis, Helsinki University of Technology, 1997.
[25] Technical Committee T13, "Self-Monitoring, analysis and reporting technology (S.M.A.R.T.)," http://www.t13.org



Junkil Ryu recieved a B.S. degree from Pohang University of Science and Technology in 2002. He is currently studying for a Ph.D. degree at the Department of Computer Science and Engineering, Pohang University of Science and Technology. His research interests include highly reliable storage systems, NAND flash memory file systems, and solid state disks.

Chanik Park received a B.E. degree in 1983 from Seoul National University, Seoul, Korea, an M.S. degree in 1985, and a Ph.D. degree in 1988, both from Korea Advanced Institute of Science and Technology, Taejeon, Korea. Since 1989, he has been working for Pohang University of Science and Technology, where he is currently a Professor with the Department of Computer Science and Engineering. He was a visiting scholar with Parallel Systems group in the IBM Thomas J. Watson Research Center in 1991, and a visiting professor with the Storage Systems group in the IBM Almaden Research Center in 1999. He served a number of international conferences as a member of Program Committee. His research interests include storage systems, embedded systems, and pervasive computing.