An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers

Shujie Han and Patrick P. C. Lee, *The Chinese University of Hong Kong;*
Fan Xu, Yi Liu, Cheng He, and Jiongzhou Liu, *Alibaba Group*

https://www.usenix.org/conference/fast21/presentation/han

This paper is included in the Proceedings of the
19th USENIX Conference on File and Storage Technologies.

February 23–25, 2021

978-1-939133-20-5

Open access to the Proceedings
of the 19th USENIX Conference on
File and Storage Technologies
is sponsored by USENIX.

An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers

Shujie Han[1], Patrick P. C. Lee[1], Fan Xu[2], Yi Liu[2], Cheng He[2], and Jiongzhou Liu[2]
[1]The Chinese University of Hong Kong [2]Alibaba Group

Abstract

Flash-based solid-state drives (SSDs) are increasingly adopted as the mainstream storage media in modern data centers. However, little is known about how SSD failures in the field are correlated, both spatially and temporally. We argue that characterizing correlated failures of SSDs is critical, especially for guiding the design of redundancy protection for high storage reliability. We present an in-depth data-driven analysis on the correlated failures in the SSD-based data centers at Alibaba. We study nearly one million SSDs of 11 drive models based on a dataset of SMART logs, trouble tickets, physical locations, and applications. We show that correlated failures in the same node or rack are common, and study the possible impacting factors on those correlated failures. We also evaluate via trace-driven simulation how various redundancy schemes affect the storage reliability under correlated failures. To this end, we report 15 findings. Our dataset and source code are now released for public use.

1 Introduction

Maintaining high storage reliability is undoubtedly important for modern data centers, yet it is often challenged by *correlated failures*, such as bursts of latent sector errors [25], correlated disk failures [5, 26], co-occurring node failures [5, 8, 11, 19], or correlated crashes of data and protocols [1]. Correlated failures complicate the design of redundancy protection schemes, which may be sufficient for tolerating independent failures but not correlated failures [19].

Modern data centers now increasingly build on flash-based solid-state drives (SSDs), and their storage reliability guarantees critically depend on the reliability of SSDs. Several field studies have characterized SSD failures in production environments, including Facebook [16], Google [2, 27], Microsoft [18], Alibaba [30], and NetApp [15] (see §6 for details). However, some of the studies [2, 15, 16, 27] analyze the proprietary customized attributes that are inapplicable for general production environments; others [18, 30] leverage the SMART (Self-Monitoring, Analysis and Reporting Technology) attributes that are known to provide statistical details for disk drive failure symptoms, yet SMART attributes do not provide the location details of how multiple failures manifest across storage *scopes* (e.g., nodes and racks). Although correlated failures are reportedly found in SSD-based data centers [15, 16], little is known about the characteristics of correlated failures and their implications on storage reliability in production environments.

To elaborate, the following questions on correlated failures remain unexplored: (i) How far are SSD failures spaced apart across different scopes in large-scale data centers? (ii) How likely does an SSD fail after another failure occurs in the same scope? (iii) How long is the time interval between two consecutive SSD failures in the same scope? (iv) Do SSD failures that are close in space imply that they are also close in time? (v) What are the factors that affect the correlated failures? (vi) What should be the proper redundancy protection schemes in the presence of correlated failures? The answers to these questions can provide insights into achieving high storage reliability in production environments.

In this paper, we present an in-depth data-driven analysis on the correlated failures, from both spatial and temporal perspectives, of SSD-based data centers at Alibaba, one of the largest Internet companies in the world. We present an extensive study on the correlated failures of nearly one million SSDs, belonging to 11 drive models from three vendors, over a span of two years. Our dataset covers the SMART logs, trouble tickets, physical locations of SSDs (e.g., nodes and racks), and the applications hosted by the underlying SSDs. Our analysis makes the following findings:

- We characterize two main types of correlated failures in the same node and rack that occur within a short time (e.g., 30 minutes), referred to as *intra-node failures* and *intra-rack failures*, respectively. We observe a non-negligible fraction of intra-node and intra-rack failures, implying the existence of strong spatial and temporal correlations of SSD failures.
- We analyze four impacting factors of drive characteristics on the correlated failures: drive models, lithography, age, and capacity. We show that such factors pose different effects on the spatial and temporal correlations of SSD failures. In particular, intra-node (intra-rack) failures likely occur in the nodes (racks) that are attached by many SSDs of the same drive model. Both intra-node and intra-rack failures of aged SSDs tend to occur within a short time.
- We analyze the impact of SMART attributes and applications on both intra-node and intra-rack failures. We find that SMART attributes have limited correlations with both intra-node and intra-rack failures and are not good indicators for detecting the existence of intra-node and intra-rack failures. Also, write-dominant applications lead to more intra-node and intra-rack failures than read-dominant ones.
- We conduct trace-driven simulation using our dataset on the impact of different redundancy protection schemes on

storage reliability. We show that redundancy schemes with high fault tolerance are critical to storage reliability under correlated failures.

We release our dataset, including the SMART logs of all failed SSDs, trouble tickets, locations, and applications, for the 11 drive models at **https://github.com/alibaba-edu/dcbrain/tree/master/ssd_open_data**. The community can leverage our dataset and findings to design effective reliability solutions in production environments. We also open-source our analysis scripts and simulator prototype at **http://adslab.cse.cuhk.edu.hk/software/ssdanalysis**.

2 Dataset

In this section, we introduce the dataset for our analysis. We describe our data collection methodology (§2.1) and study the drive population and characteristics of our dataset (§2.2). We also discuss the limitations of our dataset (§2.3).

2.1 Data Collection

We collected data from multiple SSD-based data centers at Alibaba. Each data center comprises multiple *racks*, each of which holds multiple machines called *nodes*. Each node is further attached with one or multiple SSDs.

Our dataset spans two years from January 2018 to December 2019. It covers a population of nearly one million SSDs of 11 drive models from three vendors. The SSDs are deployed in 200 K nodes of 30 K racks. Note that the SSDs of the same drive model were typically purchased from multiple batches at different times, and the SSDs attached to each node may be heterogeneous in terms of vendors, models, capacities, and deployment times. However, among the nodes with at least two SSDs, 88.6% of them are attached to the SSDs of the same drive model.

Our dataset includes multiple data types: SMART logs, trouble tickets, locations, and applications.

SMART logs. SMART is a widely adopted tool for monitoring disk drive status. It periodically reports the numerical values of the performance and reliability statistics on different dimensions, called *attributes*. Each SMART attribute includes both the raw and normalized values. Our dataset contains daily collected SMART logs over the two-year span, and its collected SMART attributes are summarized in Table 1. Since the definitions of SMART attributes vary across vendors, for easy comparison, we focus on the SMART attributes that are reported by more than half of SSDs (shown in the "Reported%" column). We categorize the SMART attributes by their monitoring types into five groups, namely internal errors, spare blocks, wearout degree, workload, and power. Some SMART attributes have identical meanings but are assigned different SMART IDs by vendors (e.g., S170/S180, S171/S181, and S172/S182). Also, some SMART attributes have vendor-specific raw values (marked with an asterisk '*' in Table 1), so we only consider their normalized values.

Category	ID	Attribute name	Reported%
Internal errors	S5	Reallocated sector count	100.0%
	S183	SATA downshift error count	96.5%
	S184	End-to-end errors	100.0%
	S187	Reported uncorrectable errors	100.0%
	S195	Hardware ECC recovered	55.4%
	S197	Current pending sector count	87.5%
	S199	UltraDMA CRC error count	100.0%
	S171/S181	Program failed count	100.0%
	S172/S182	Erase failed count	100.0%
Spare blocks*	S170/S180	Available reserved blocks	100.0%
Wearout degree*	S173	Wear leveling count	100.0%
	S177	Wear range delta	
	S233	Media wearout indicator	
Workload	S241	Number of blocks written	68.8%
	S242	Number of blocks read	56.3%
Power	S9	Power on hours	100.0%
	S12	Power cycle count	99.1%
	S174	Unexpected power loss count	78.5%
	S175*	Power loss protection failure	57.0%

Table 1: Overview of SMART attributes in our dataset. "Reported%" is the percentage of SSDs with the corresponding SMART attribute. Only the normalized values are considered for the vendor-specific SMART attributes marked by an asterisk '*'.

Trouble tickets. Each node runs a background monitoring daemon that periodically collects SMART statistics and system-level logs/alerts from its attached SSDs and sends the collected data to a centralized maintenance system that monitors failures. The maintenance system applies rule-based detection, defined by administrators, to detect and report any failure behavior in the form of *trouble tickets*. Each trouble ticket records the node ID, drive ID, timestamp, and failure description. Administrators further manually validate each trouble ticket to confirm the failure status. We use the trouble tickets as the ground-truths for our failure analysis. Throughout the two-year span, we collected about 19 K trouble tickets (i.e., 19 K failed SSDs in total).

Our trouble tickets cover two main types of SSD failures: (i) *whole drive failures*, in which an SSD either cannot be accessed or loses all data that is unrecoverable; and (ii) *partial drive failure*, in which part of the data in an SSD either cannot be accessed and is unrecoverable.

Locations. Our dataset records the physical location of each SSD, including the machine room ID, rack ID, node ID, drive ID, and slot number. In particular, we can correlate an SSD to the SMART logs and trouble tickets by its drive ID.

Applications. Our dataset covers hundreds of applications, including both internal (e.g., resource management, development, testing, etc.) and external services (e.g., web services, data analytics, etc.). Each node is configured to serve a single application (note that the applications within a rack may be different) and distributes a set of tasks to the attached SSDs as evenly as possible. We can correlate an SSD to its hosted

Applications	Total%	Failures%
Web service management (WSM)	39.4%	48.5%
Resource management (RM)	19.1%	16.4%
Web proxy services (WPS)	4.6%	2.9%
SQL services (SS)	3.4%	1.0%
Database (DB)	2.8%	1.1%
Web services (WS)	1.8%	1.3%
Data analytics engine (DAE)	1.7%	6.6%
Network attached storage (NAS)	1.5%	2.9%

Table 2: Overview of the top eight most widely used applications with more than hundreds of failed SSDs, including the percentage of deployed SSDs in the whole population ("Total%") and the percentage of SSD failures in the failed SSD population ("Failures%"). Note that SS and DB are two similar applications, but belong to different business units.

application via its node ID. Table 2 shows the top eight most widely used applications, each of which contains hundreds of failed SSDs in our dataset. Specifically, WSM covers 39.4% of all SSDs and 48.5% of all failed SSDs. WPS, SS, and DB cover 10.8% of all SSDs, while covering only 5.0% of all failed SSDs. DAE and NAS have 3.2% of all SSDs, while covering 9.5% of all failed SSDs. We will give a detailed analysis on the relationships between the failure patterns and workload distributions of the eight applications (§4.4).

2.2 Summary of Statistics

We first analyze the basic statistics and SSD characteristics in our dataset, as shown in Table 3.

Population statistics. We consider 11 drive models from three vendors. Each drive model is denoted by "Vendor""*k*", where "Vendor" is represented by a letter ('A', 'B', and 'C') for each of the three vendors, and "*k*" (1 to 6) refers to the *k*-th most numerous model in the same vendor. The first three columns in Table 3 show the percentages of each drive model in the same vendor and the whole population. The 11 drive models together cover nearly one million SSDs.

Drive characteristics. The fourth to sixth columns in Table 3 describe the key drive characteristics, including the flash technology, lithography, and capacity. All 11 drive models use the SATA interface. The drive models in vendors A and B build on enterprise-class MLC NAND cells, while those in vendor C build on 3D-TLC flash. These drive models have different lithography parameters (bill-of-material (BOM) revision for 3D-TLC) and capacities (ranging from 240 to 1920 GB).

Usage. The seventh to ninth columns in Table 3 show the statistical summaries of SSD usage, including the over-provisioning (OP) factor (i.e., the fraction of dedicatedly reserved space in SSDs for internal garbage collection), the average power-on years computed from S9 (Table 1), and the mean of rated life used (i.e., the percentage of erase cycles over the erase cycle limit) computed from the SMART attributes related to the wearout degree (Table 1).

Reliability. The last three columns in Table 3 show three reliability metrics, including the mean percentage of spare blocks used, the mean number of bad sectors, and the annualized failure rate (AFR). We compute the percentage of spare blocks using the SMART attributes related to spare blocks (S170/S180), and the number of bad sectors using S5 in Table 1. We define the AFR by the following formula [13, 18]:

$$AFR(\%) = \frac{f}{n_1 + n_2 + \ldots + n_{two-year}} \times 365 \times 100,$$

where f is the total number of failed SSDs reported in our trouble tickets and n_i is the number of operational SSDs on day i over the two-year span. The overall AFR of all MLC SSDs (A1 to A6 and B1 to B3) is 0.55%, and their AFRs range from 0.16% to 2.52%, slightly lower than those reported for SSDs in Google's data centers (1-2.5%) [27]. In contrast, the AFRs of 3D-TLC SSDs (C1 and C2) are higher than 3%. The overall AFR of all SSDs in our dataset is 1.16%.

2.3 Limitations

Our analysis has the following limitations, mainly due to the unavailable information in our dataset.

Data missing. We expect that the SMART logs contain daily statistics without loss, yet our dataset indeed contains incomplete SMART data over time in both failed and healthy SSDs. Reasons of such data missing include network failures, software maintenance or upgrades, system crashes, etc. In this work, we mainly focus on analyzing the correlations of SSD failures via trouble tickets, rather than the correlations of SMART attributes over time. Thus, the data missing in the SMART logs does not compromise our analysis.

Failure symptoms. Our dataset reports SSD failures via trouble tickets, but does not cover the failure symptoms at the operating system level. Such failure symptoms can be found in kernel syslogs, which are not collected in our dataset.

Drive repair. Our dataset does not include the repair details for failed SSDs. In practice, how long the data in a failed SSD is recovered depends on the importance of its stored data to the upper-level applications. Administrators may not immediately repair the failed SSDs that store less critical data to save operational overhead [2, 30]. Due to limited details, we assume that all SSDs store data with the same importance, and the repair time depends on the amount of data to be reconstructed (§5).

Redundancy protection. Production storage systems use erasure coding for redundancy protection against failures [8, 12, 17]. In Alibaba production, 3-way replication is the commonly used redundancy mechanism [30]. However, the redundancy parameters may also vary across applications and we do not have access to the redundancy parameters for each application. In this work, we assume that all applications adopt the same redundancy parameters to drive our reliability analysis (§5).

Population statistics			Drive characteristics			Usage			Reliability		
Model	Vendor%	Total%	Flash Tech.	Lithography	Capacity	OP	Power-on years	Rated life used (%)	Spare blocks used (%)	# bad sectors	AFR (%)
A1	52.3%	29.8%	MLC	20 nm	480 GB	7%	4.6	17.8 (±0.067)	0.18 (±0.0080)	9.3 (±0.60)	0.16%
A2	21.8%	12.4%	MLC	20 nm	800 GB	28%	4.5	17.2 (± 0.15)	0.19 (±0.013)	12.5 (±1.3)	0.46%
A3	7.9%	4.5%	MLC	20 nm	480 GB	7%	5.5	25.9 (±0.41)	0.022 (±0.012)	12.4 (±2.4)	2.36%
A4	7.2%	4.1%	MLC	16 nm	240 GB	7%	3.2	8.8 (±0.074)	0.064 (±0.013)	2.4 (±0.72)	0.64%
A5	5.7%	3.3%	MLC	16 nm	480 GB	7%	3.2	27.0 (±0.28)	0.087 (±0.015)	5.0 (±1.2)	0.45%
A6	5.1%	2.9%	MLC	20 nm	800 GB	28%	4.6	24.7 (±0.44)	0.018 (±0.013)	13.7 (±2.9)	0.49%
B1	51.5%	10.3%	MLC	21 nm	480 GB	7%	3.8	6.4 (±0.029)	0.0063 (±0.0010)	0.036 (±0.024)	0.21%
B2	25.5%	5.1%	MLC	19 nm	1920 GB	7%	3.3	2.0 (±0.014)	0.086 (±0.0092)	12.2 (±1.4)	0.71%
B3	23.0%	4.6%	MLC	24 nm	1920 GB	7%	2.1	3.6 (±0.028)	0.021 (±0.0041)	0.50 (±0.25)	2.52%
C1	89.3%	20.6%	3D-TLC	V1	1920 GB	7%	2.0	4.3 (±0.022)	0.064 (±0.0054)	10.1 (±0.74)	3.29%
C2	10.7%	2.5%	3D-TLC	V1	960 GB	7%	1.4	2.0 (±0.062)	0.0049 (±0.0047)	0.67 (±0.37)	3.92%

Table 3: Summary of statistics of collected dataset. The population statistics include the percentage of drives in the same vendor ("Vendor%") and the percentage of drives in the whole drive population in the dataset ("Total%"). For the "Rated life used", "Spare blocks used", and "# bad sectors" columns, each value in brackets denotes the 95% confidence interval.

3 Overview of Analysis Methodology

Our analysis studies the correlated failures of SSDs in our dataset, and focuses on several dimensions.

Spatial and temporal properties. We study how SSD failures manifest within a scope, either a node or a rack, within a certain time period. We consider both *intra-node failures* and *intra-rack failures* to refer to the failures co-occurring within a node and a rack, respectively. We also define the *intra-node (intra-rack) failure time interval* as the time interval between two consecutive failures that co-occur within the same node (rack). We refer to a failure as an intra-node (intra-rack) failure if its intra-node (intra-rack) failure time interval with its preceding or following failure in the same node (rack) is smaller than a pre-specified threshold. Here, we set a default threshold as 30 minutes, assuming that this is the minimum time for a failure to be detected before it is repaired [12]. In other words, a node (rack) may contain more than one active failure at a time under intra-node (intra-rack) failures. We define the *intra-node (intra-rack) failure group* as a sequence of intra-node (intra-rack) failures starting from an intra-node (intra-rack) failure without a preceding one until an intra-node (intra-rack) failure without a following one. We also vary the thresholds of the intra-node and intra-rack failure time interval in our analysis.

Correlation properties. We use the Spearman's Rank Correlation Coefficient (SRCC) [29] to measure the correlation of two variables. For example, to measure the correlation between an SSD failure and a SMART attribute using the SRCC, we use an indicator variable to represent if an SSD is failed (i.e., 1 means failed; or 0 otherwise), and a numerical variable to represent the value of a SMART attribute. The SRCC calculates the Pearson Correlation Coefficient [21] between the rank values of two variables to measure their monotonic relationships. The SRCC ranges from -1 (i.e., high negative correlation) to +1 (i.e., high positive correlation); a zero SRCC means that the two variables are independent.

4 Correlation Analysis

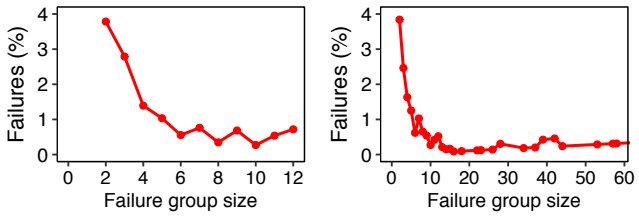
We analyze the correlated failures of SSDs in our dataset in four aspects: (i) spatial and temporal correlations among failures (§4.1), (ii) the impacting factors on correlated failures, including the drive models, lithography, age, and capacity (§4.2), (iii) the impact of SMART attributes on correlated failures (§4.3), and (iv) the impact of applications on correlated failures (§4.4). Finally, we discuss the implications of our findings (§4.5).

4.1 Correlations among Failures

We first examine the severity of correlated failures by the intra-node and intra-rack failure group sizes (i.e., by counting the number of failures within a group). Figure 1 shows the percentage of failures versus the intra-node or intra-rack failure group sizes; note that for Figure 1(b), we omit the plots for the intra-rack failure group sizes that exceed 60 (the maximum is 89) due to the sparseness. We see that a non-negligible fraction of SSD failures belong to intra-node and intra-rack failures. In particular, 12.9% (18.3%) of failures are intra-node (intra-rack) failures. Also, the intra-node and intra-rack failure group size can exceed the tolerable limit of some typical redundancy protection schemes (e.g., four failures) (see §5 for details).

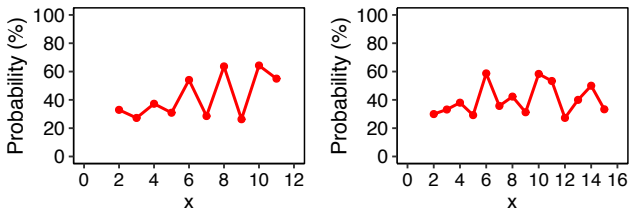
Finding 1. *A non-negligible fraction of SSD failures belong to intra-node and intra-rack failures (12.9% and 18.3% in our dataset, respectively). Also, the intra-node and intra-rack failure group size can exceed the tolerable limit of some typical redundancy protection schemes.*

We further check whether the likelihood of an SSD failure depends on the already existing SSD failures among the intra-node and intra-rack failures. Borrowing the idea by Mesa et al. [16], we compute the conditional probability of having an additional SSD failure per intra-node (intra-rack) failure group given the existing intra-node (intra-rack) failures, by dividing the number of intra-node (intra-rack) failure groups



(a) Intra-node failures (b) Intra-rack failures

Figure 1: Finding 1. Percentages of failures for different intra-node and intra-rack failure group sizes.



(a) Intra-node failures (b) Intra-rack failures

Figure 2: Finding 2. Conditional probabilities of having an additional SSD failure for different failure group sizes per intra-node or intra-rack failure group.

with a failure group size of $x+1$ by the number of intra-node (intra-rack) failure groups with a failure group size of x or $x+1$.

Figure 2 shows that the conditional probability of an additional SSD failure depends on the already existing SSD failures among the intra-node and intra-rack failures. The conditional probability of having an additional SSD failure in an intra-node (intra-rack) failure group ranges from 26.3% to 64.3% as x ranges from 2 to 11 (from 27.3% to 58.7% as x ranges from 2 to 88); note that we omit the plots for the intra-rack failure group size that exceeds 16 in Figure 2(b) due to the sparseness. If there is no correlation among intra-node (intra-rack) failures and the SSD failures are uniformly distributed on nodes (racks), the conditional probability of having an additional SSD failure given the existing intra-node (intra-rack) failures is similar to the AFR [16].

Finding 2. *The likelihood of having an additional intra-node (intra-rack) failure in an intra-node (intra-rack) failure group depends on the already existing intra-node (intra-rack) failures.*

We examine how the percentages of intra-node and intra-rack failures are affected by various thresholds of the intra-node and intra-rack failure time intervals, respectively. Figure 3 shows that a non-negligible fraction of intra-node and intra-rack failures occur within a short period of time. The intra-node (intra-rack) failures with one month as the threshold of the failure time interval account for 29.2% (63.0%). When the threshold of the failure time interval falls in one minute, the intra-node (intra-rack) failures still account for 10.0% (14.4%).

Finding 3. *A non-negligible fraction of intra-node and intra-rack failures occur within a short period of time, even within one minute.*

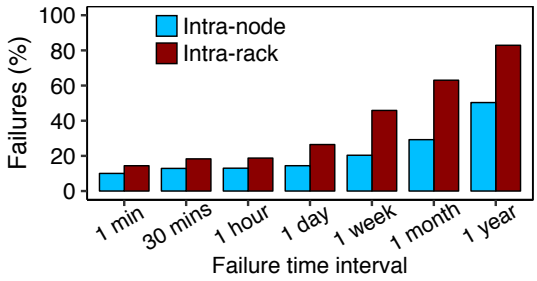


Figure 3: Finding 3. Percentages of intra-node (intra-rack) failures broken down by different thresholds of the intra-node (intra-rack) failure time intervals.

4.2 Impacting Factors on Correlated Failures

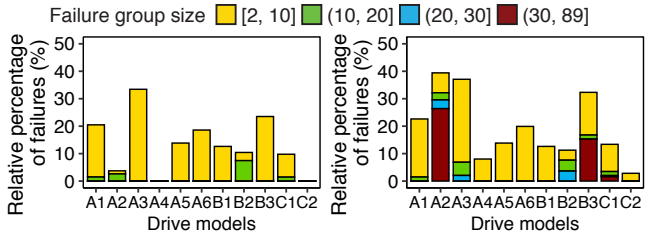
We next study how various factors affect the spatial and temporal correlations of failures.

4.2.1 Drive Models

We analyze the impact of drive models on correlated failures. Figure 4 shows that the relative percentages of failures (over all SSD failures of the same drive model) for different sets of intra-node and intra-rack failure group sizes vary highly across the drive models. In particular, the relative percentages of intra-node (intra-rack) failures range from 0% to 33.4% (from 2.8% to 39.4%). Interestingly, A2 has only 3.7% of intra-node failures, but has 39.4% of intra-rack failures, among which 26.4% reside in the intra-rack failure groups of sizes larger than 30.

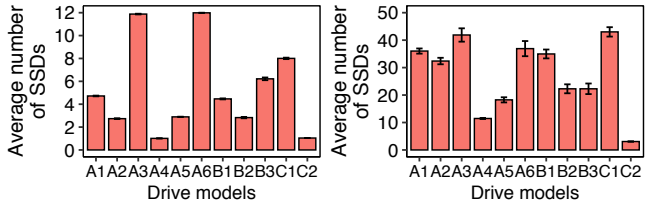
We next examine the reason of high percentages of intra-node and intra-rack failures of some drive models, by examining the average numbers of SSDs per node or rack for different drive models. Figure 5 shows the distribution of the average number of SSDs per node and rack (each error bar shows the 95% confidence interval). In general, putting more SSDs from the same drive model in the same nodes (racks) leads to a higher percentage of intra-node (intra-rack) failures.

However, we observe some exceptions. A3 and A6 have the same average number of SSDs per node (i.e., 12.0), but the relative percentage of intra-node failures for A3 is higher than that for A6 by 14.8%. One possible reason is that the AFR of A3 (2.36%) is 5× that of A6 (0.49%). Note that the AFR is not always the root cause of leading to high relative percentages of intra-node and intra-rack failures. For example, one exception is that C1 has more average number of SSDs per node (rack) than B3 by 1.8 (20.7), but the relative percentage of intra-node (intra-rack) failures for C1 is lower than that for B3 by 13.7% (18.9%). Similar exceptions include the intra-rack failures for A2 and A3. However, the AFR of B3 (A2) is lower than that of C1 (A3) by 0.77% (1.9%). We further examine the machine rooms where intra-rack failures reside for A2 and B3. We observe that the relative percentages of intra-rack failures



(a) Intra-node failures (b) Intra-rack failures

Figure 4: Finding 4. Relative percentages of failures for different sets of intra-node or intra-rack failure group sizes across the drive models.



(a) Per node (b) Per rack

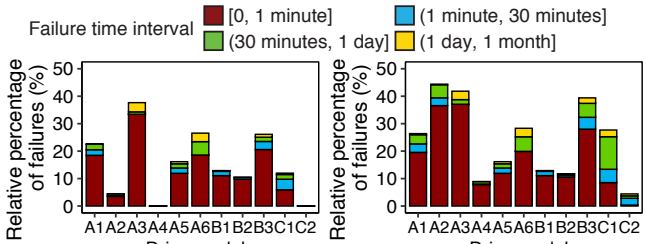
Figure 5: Finding 4. Average numbers of SSDs per node or rack for the drive models (with 95% confidence intervals as error bars).

from two machine rooms account for 29.6% and 15.3% for A2 and B3, respectively, and the intra-rack failure group sizes are larger than 20 and 30 for A2 and B3, respectively. Thus, the high relative percentages of intra-rack failures may also be attributed to the machine rooms (e.g., high temperature in a machine room can lead to more SSD failures [30]).

Finding 4. *The relative percentages of intra-node and intra-rack failures vary across drive models. Putting too many SSDs from the same drive model in the same nodes (racks) leads to a high percentage of intra-node (intra-rack) failures. Also, the AFR and environmental factors (e.g., temperature) affect the relative percentages of intra-node and intra-rack failures.*

We vary the thresholds of the intra-node and intra-rack failure time intervals, broken down by the drive models. Figure 6 shows that the intra-node and intra-rack failures with a short failure time interval account for non-negligible percentages for most drive models. In particular, the relative percentages of intra-node (intra-rack) failures with a threshold of one day range from 4.4 to 34.3% (from 11.8 to 44.2%), except for A4 and C2 due to their limited numbers of SSDs per node or rack. The relative percentages of intra-node (intra-rack) failures with a threshold of one minute still account for 3.5-33.4% (7.8-37.1%) except for A4 and C2 (C2).

Finding 5. *There exist non-negligible fractions of intra-node and intra-rack failures with a short failure time interval for most drive models (e.g., up to 33.4% and 37.1% with a failure time interval of within one minute in our dataset, respectively) .*



(a) Intra-node failures (b) Intra-rack failures

Figure 6: Finding 5. Relative percentages of failures for different thresholds of intra-node or intra-rack failure time intervals across the drive models.

4.2.2 Lithography

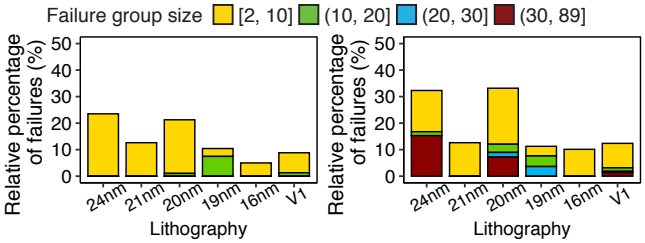
We analyze the impact of lithography on correlated failures. For MLC SSDs, a smaller lithography implies that the SSDs have a higher density. Also, 3D-TLC SSDs (C1 and C2) have higher densities than those of the MLC SSDs. Figure 7 shows that the SSDs of a smaller lithography (i.e., a higher density) generally have lower relative percentages of intra-node and intra-rack failures (over all SSD failures of the same lithography). In particular, for MLC SSDs, the relative percentages of intra-node (intra-rack) failures decrease from 23.5% to 5.0% (from 32.3% to 10.1%) from 24 nm to 16 nm. An exception is 21 nm SSDs, due to its limited number of failures. For 3D-TLC SSDs, the relative percentages of both intra-node and intra-rack failures are close to 19 nm MLC SSDs.

We also vary the thresholds of the intra-node and intra-rack failure time intervals, broken down by the lithography. Figure 8 shows that the relative percentages of intra-node and intra-rack failures for different thresholds decrease generally with a smaller lithography for MLC SSDs. In particular, the relative percentages of intra-node (intra-rack) failures with a threshold of one minute increase from 20.6% to 4.3% (28.0% to 9.3%) from 24 nm to 16 nm except for 21 nm SSDs due to few failures. The intra-rack failures with a threshold of one minute for 20 nm and 24 nm SSDs account for higher percentages than other MLC SSDs by 18.7-22.4%, since they include the intra-rack failures from A2 and B3, respectively (Figure 4(b)).

Finding 6. *MLC SSDs with higher densities generally have lower relative percentages of intra-node and intra-rack failures.*

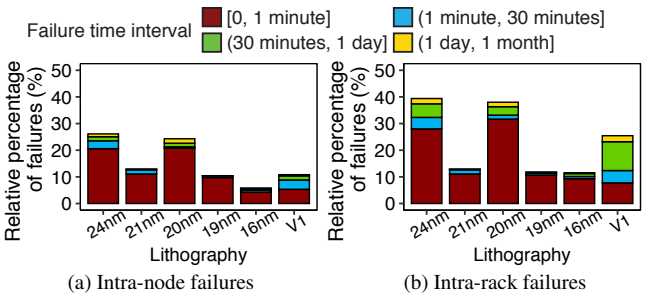
4.2.3 Age

We analyze the impact of the age of a failed SSD (e.g., the power-on years until the failure occurs) on correlated failures. Figure 9 shows that the relative percentages of intra-node (intra-rack) failures (over all SSD failures of the same age group) for different sets of intra-node (intra-rack) group sizes increase with age in general. In particular, the relative percentages of intra-node (intra-rack) failures of each age group increase from 6.8% to 33.2% (from 11.0% to 37.6%) from



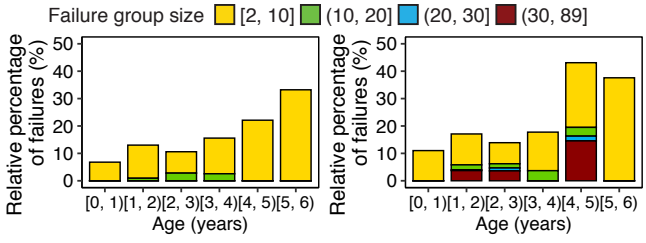
(a) Intra-node failures (b) Intra-rack failures

Figure 7: Finding 6. Relative percentages of failures for different sets of intra-node or intra-rack failure group sizes across the lithography.



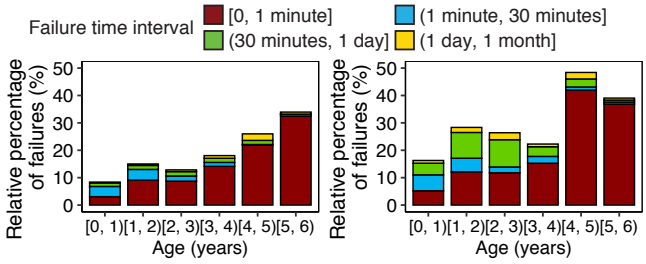
(a) Intra-node failures (b) Intra-rack failures

Figure 8: Finding 6. Relative percentages of failures for different thresholds of intra-node or intra-rack failure time intervals across the lithography.



(a) Intra-node failures (b) Intra-rack failures

Figure 9: Finding 7. Relative percentages of failures for different sets of intra-node or intra-rack failure group sizes across the age.



(a) Intra-node failures (b) Intra-rack failures

Figure 10: Finding 7. Relative percentages of failures for different thresholds of intra-node or intra-rack failure time intervals across the age.

zero to six years old. Also, the relative percentage of intra-node (intra-rack) failures for 1-2 years old is slightly higher than that for 2-3 years old by 2.4% (3.2%). One possible reason is that the infant mortality of SSD failures can last for more than a year [15].

Figure 10 shows a noticeable trend that the intra-node and intra-rack failures at an older age are more likely to occur within a short time. In particular, the relative percentages of intra-node (intra-rack) failures with a threshold of one minute increase from 3.1% to 32.5% (from 5.2% to 36.9%) from zero to six years old. We also examine the average rated life used for intra-node and intra-rack failures at different ages (not shown in plots). The rated life used for intra-node (intra-rack) failures (with the default threshold of 30 minutes) increases from 1.6% (1.4%) for 0-1 year old to 67.5% (68.4%) for 5-6 years old on average, showing that a longer rated life used increases the likelihood of intra-node and intra-rack failures.

Finding 7. *The relative percentages of intra-node and intra-rack failures increase with age. The intra-node and intra-rack failures at an older age are more likely to occur within a short time due to the increasing rated life used.*

4.2.4 Capacity

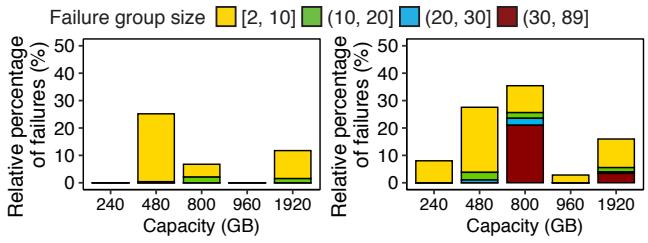
We examine the impact of the capacity on correlated failures. Figure 11 shows that the relative percentages of intra-node (intra-rack) failures (over all SSD failures of the same capacity) for different sets of intra-node (intra-rack) failure group sizes vary significantly across the capacity. Specifically, the relative percentages of intra-node (intra-rack) failures for each capacity range from 0% to 25.2% (from 2.9% to 35.4%). As the SSDs with capacities of 480 GB, 800 GB, and 1920 GB cover more failures (Table 3), they have higher relative percentages of intra-node and intra-rack failures.

We next vary the thresholds of the intra-node and intra-rack failure time intervals, broken down by the capacity. Figure 12 shows no clear trend between the relative percentages of intra-node or intra-rack failures and the capacity. In particular, the 480 GB SSDs have the highest relative percentage of intra-node failures with a threshold of one minute, since they cover A3 with 34.4% of intra-node failures (Figure 6(a)), while the 800 GB SSDs have the highest relative percentage of intra-rack failures with a threshold of one minute, since they cover A2 with 36.6% of intra-rack failures (Figure 6(b)).

Finding 8. *The relative percentages of intra-node and intra-rack failures vary significantly across the capacity. There is no clear trend between the relative percentages of intra-node (or intra-rack) failures for different thresholds of failure time intervals and the capacity.*

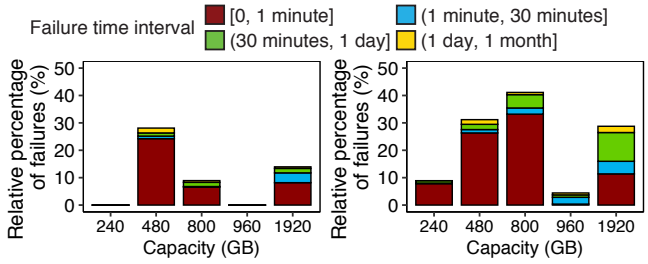
4.3 Impact of SMART Attributes

We analyze how SMART attributes are correlated with intra-node and intra-rack failures. We use the SRCC [29] (§3) to examine which SMART attributes are correlated with intra-node and intra-rack failures. Figure 13 shows that the SMART attributes have limited correlations with intra-node or intra-rack failures, and the differences of the absolute values of SRCC between intra-node and intra-rack failures are very



(a) Intra-node failures (b) Intra-rack failures

Figure 11: Finding 8. Relative percentages of failures for different sets of intra-node or intra-rack failure group sizes across the capacity.



(a) Intra-node failures (b) Intra-rack failures

Figure 12: Finding 8. Relative percentages of failures for different thresholds of intra-node or intra-rack failure time intervals across the capacity.

small. In particular, the SMART attributes related to internal errors (e.g., S187) are more correlated with intra-node and intra-rack failures, yet the highest SRCC values are only 0.23 for both intra-node and intra-rack failures. This implies that SMART attributes are not good indicators for detecting the existence of intra-node and intra-rack failures. Furthermore, the differences of the absolute values of SRCC between intra-node and intra-rack failures are very small and less than 0.02.

Finding 9. *The SMART attributes have limited correlations with intra-node and intra-rack failures, and the highest SRCC values (from S187) are only 0.23 for both intra-node and intra-rack failures. Thus, SMART attributes are not good indicators for detecting the existence of intra-node and intra-rack failures. Also, intra-node and intra-rack failures have no significant difference of the absolute values of SRCC for each SMART attribute.*

4.4 Impact of Applications

We analyze the relationships between the failure patterns and workload distributions of the eight applications (Table 2), and study the impact of applications on correlated failures.

We first examine the relationships between the AFRs and workload distributions of the eight applications. In particular, we use the raw values of SMART attributes S241 and S242 to calculate the percentage of writes among the total workloads of reads and writes, and determine if each SSD is read-dominant (i.e., more reads than writes) or write-dominant (i.e., more writes than reads). Figure 14(a) shows the average percentages of writes per SSD for the eight applications (each

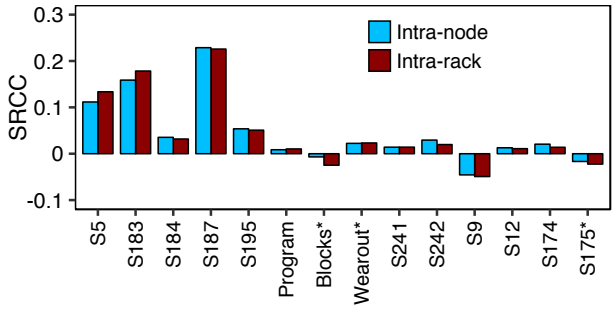
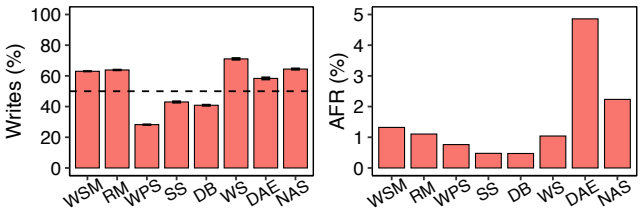


Figure 13: Finding 9. SRCC values between each SMART attribute and intra-node or intra-rack failures. Note that we omit three SMART attributes, including S197, S199, and erase failed counts, since their absolute SRCC values are less than 0.01 for both intra-node and intra-rack failures.



(a) Average percentages of writes per SSD (b) AFRs

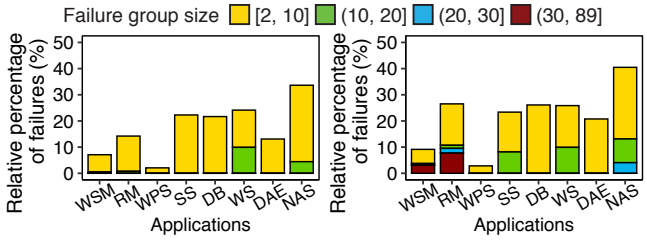
Figure 14: Finding 10. Average percentages of writes per SSD (with 95% confidence intervals as error bars) and AFRs for the applications.

error bar shows the 95% confidence interval). Reads are dominant for WPS, SS, and DB, while writes are dominant for the remaining five applications. Figure 14(b) shows the AFRs for the applications. The AFRs of write-dominant applications in general are higher than those of read-dominant applications. This implies that write-dominant workloads lead to more SSD failures overall, conforming to prior findings [18].

However, write-dominant workloads are not the only impacting factor on the AFRs. We see that DAE has the highest AFR (i.e., 4.9%), and it is mainly hosted on the drive model C1, which has a high AFR (3.29% in Table 3). Also, WPS has a higher AFR than SS and DB by 0.29%, although it has a lower percentage of writes than SS and DB. The reason is that C1 is mainly used in WPS, while A1, which has a low AFR (0.16% in Table 3), is the drive model mainly used in SS and DB.

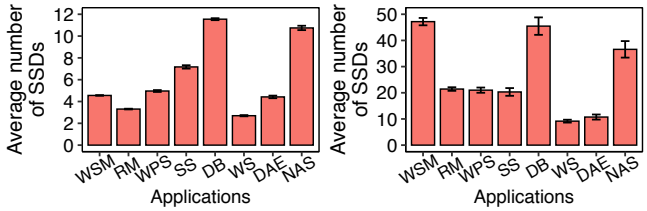
Finding 10. *Write-dominant workloads lead to more SSD failures overall, but are not the only impacting factor on the AFRs. Other factors (e.g., drive models) can affect the AFRs.*

We analyze the impact of applications on correlated failures. Figure 15 shows that the relative percentages of intra-node (intra-rack) failures (over all SSD failures of the same application) for different sets of intra-node (intra-rack) failure group sizes vary across the applications. In particular, the relative percentages of intra-node (intra-rack) failures for the applications range from 2.1% to 33.6% (from 2.8% to 40.5%).



(a) Intra-node failures (b) Intra-rack failures

Figure 15: Finding 11. Relative percentages of failures for different sets of intra-node or intra-rack failure group sizes across the applications.



(a) Per node (b) Per rack

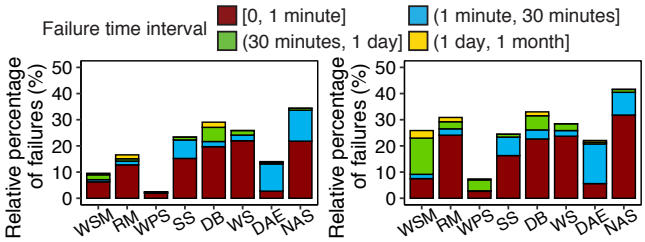
Figure 16: Finding 11. Average numbers of SSDs per node or rack for the applications (with 95% confidence intervals as error bars).

To explain these differences across the applications, we examine the average number of SSDs per node or rack for each application. Figure 16 shows that attaching more SSDs on nodes and racks for applications tends to have a high percentage of intra-node (intra-rack) failures. However, there are some exceptions. The average number of SSDs per node for WSM (4.6) is close to that of WPS (5.0), yet the relative percentage of intra-node failures for WSM is higher than that of WPS by 5.0%. The reason is that WPS has read-dominant workloads, while WSM has write-dominant workloads that lead to more failures (Figure 14(a)). Similar observations also hold for intra-rack failures. The average number of SSDs per rack for DAE (10.7) is much less than that for WPS (21.0), yet the relative percentage of intra-rack failures of DAE is higher than that of WPS by 17.9%.

Finding 11. *The applications with more SSDs per node (rack) and write-dominant workloads tend to have a high percentage of intra-node (intra-rack) failures.*

We further examine the impact of applications on correlated failures by varying the thresholds of the intra-node and intra-rack failure time intervals. Figure 17 shows that the relative percentages of intra-node and intra-rack failures for different thresholds of the failure time intervals vary across the applications. In particular, the relative percentages of intra-node (intra-rack) failures with a threshold of one minute account for 1.9-22.0% (2.6-31.8%).

To explain these differences among the applications, we examine the average ages of intra-node and intra-rack failures for the applications (not shown in plots). The average ages of intra-node (intra-rack) failures with a threshold of one minute for RM, SS, DB, and WS range from 3.2 to 3.9 years old



(a) Intra-node failures (b) Intra-rack failures

Figure 17: Finding 12. Relative percentages of failures for different thresholds of intra-node or intra-rack failure time intervals across the applications.

(from 3.2 to 4.1 years old), which are older than those of the remaining applications, i.e., from 1.3 to 2.5 years old (from 1.2 to 2.2 years old). This conforms to Finding 7. However, there are two exceptions: (i) The average ages of intra-node and intra-rack failures for WS are younger than those for SS and DB by 0.48-0.65 years, while the relative percentage of intra-node (intra-rack) failures with a threshold of one minute for WS is higher than those for SS and DB by 2.3-4.5% (1.0-7.4%). (ii) The average age of intra-node (intra-rack) failures for NAS is younger than that for WPS by 1.0 (0.79) years, while the relative percentage of intra-node (intra-rack) failures with a threshold of one minute for NAS is higher than that for WPS by 19.9% (29.1%). The reasons for these exceptions are due to more write-dominant workloads for WS and NAS (Figure 14(a)).

Finding 12. *Among individual applications, the intra-node and intra-rack failures at an older age and with more write-dominant workloads tend to occur in a short time.*

4.5 Discussion

We highlight the findings in the correlation analysis:

- Intra-node and intra-rack failures commonly exist in SSD failures. Even worse, a non-negligible fraction of intra-node and intra-rack failures occur within a short time. In the presence of intra-node and intra-rack failures, it is critical to deploy the redundancy protection schemes with high fault tolerance to cope with such correlated failures.
- We analyze the effects of the four impacting factors, namely drive models, lithography, age, and capacity, on intra-node and intra-rack failures. We find that drive models and age have larger impacts on correlated failures than lithography and capacity. Also, intra-node (intra-rack) failures tend to occur with many SSDs from the same drive model on the same node (rack), and the intra-node and intra-rack failures of aged SSDs are more likely to occur within a short time. System operators should avoid putting such SSDs in the same scope to limit the occurrences of correlated failures.
- Intra-node and intra-rack failures have limited correlations with the SMART attributes and have no significant differ-

ences of correlations with each SMART attribute. Thus, the SMART attributes are not good indicators for detecting the existence of intra-node and intra-rack failures in practice. Other data sources, such as system logs, may be useful to detect any potential correlated failures.

- In addition to SSD characteristics, applications also play a role in the behavior of correlated failures. Intra-node and intra-rack failures are more likely to occur in write-dominant applications than read-dominant ones. Thus, high fault-tolerance protection schemes are more essential for write-dominant applications.

5 Case Study: Redundancy Protection

In this section, we present a trace-driven simulation analysis on how redundancy schemes affect the storage reliability in the face of correlated failures using our dataset.

5.1 Simulation Methodology

Redundancy schemes. Replication and erasure coding are two widely adopted redundancy approaches to provide fault tolerance in modern data centers. Our analysis considers three redundancy schemes:

- *r-way replication* (Rep(r)): For each data chunk, it makes $r > 1$ exact chunk copies to tolerate any $r-1$ chunk failures. We consider Rep(2) and Rep(3), where Rep(3) is used by traditional distributed file systems [6, 9].
- *Reed-Solomon coding* [23] (RS(k,m)): For every coding group of k data chunks, it encodes them into m parity chunks, such that any k out of $k+m$ data/parity chunks (i.e., any m chunk failures can be tolerated). We consider RS(6,3) (used by Google Colossus [7] and Quantcast File System [20]), RS(10,4) (used by Facebook [17]), and RS(12,4) (the same redundancy as in Azure [12]).
- *Local Reconstruction Coding* [12] (LRC(k,l,g)): For every coding group of k data chunks, it encodes each subgroup of k/l data chunks into a local parity chunk, and encodes all k data chunks into g global parity chunks. Thus, each single chunk failure can be reconstructed from any k/l non-failed chunks, while tolerating any $g+1$ chunk failures. We consider LRC(12,2,2), as used by Azure [12]. Note that it has the same redundancy as RS(12,4), but can only tolerate any three chunk failures and some of the four chunk failures (but not all four chunk failures as in RS(12,4)).

Replication is simple to implement, but incurs high storage overhead. Reed-Solomon coding incurs much lower storage overhead than replication, but incurs high *repair bandwidth* since any lost chunk needs to be reconstructed by accessing k non-failed chunks. Local Reconstruction Coding mitigates the repair bandwidth as any lost chunk can now be reconstructed by k/l non-failed chunks.

To mitigate repair bandwidth, we also consider *lazy recovery* [28], which triggers a repair operation only when more than one chunk fails (in Reed-Solomon coding, all data chunks remain available if no more than m chunks fail). This is in contrast to *eager recovery*, which triggers a repair operation immediately when there exists any failed chunk.

Simulator. We extend the C++ discrete-event simulator SIMEDC [31] to support the reliability evaluation on our dataset. Our simulator runs multiple iterations. In each iteration, it initializes the data center topology, redundancy scheme, and chunk placement. It issues the failure events based on the chronological failure patterns in our dataset. It also generates the repair events, whose repair durations depend on the amount of repair bandwidth and the available data center capacity; for lazy recovery, the repair events are triggered only when a threshold number of failures occurs. Each iteration runs over a mission time. To generate randomness across iterations, we configure random chunk placements (see details below). We report the averaged results over all iterations.

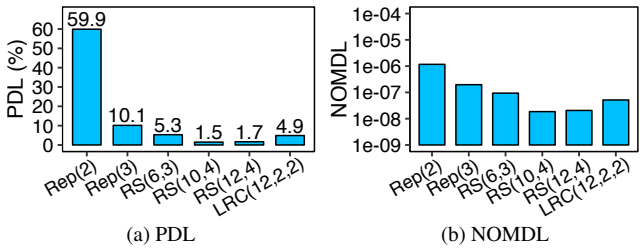
Metrics. We measure the reliability with the following metrics over the mission time:

- *Probability of data loss (PDL)*. It measures the likelihood that (unrecoverable) data loss occurs in a data center (i.e., the number of chunk failures in a coding group exceeds the tolerable limit).
- *Normalized magnitude of data loss (NOMDL)* [10]. It measures the amount of (unrecoverable) data loss (in bytes) normalized to the storage capacity.

Simulator setup. We configure the chunks in a coding group to be stored on different racks (one chunk per rack), so as to provide both node-level and rack-level fault tolerance. However, in our dataset, the number of racks varies highly across the clusters. Thus, we focus on the clusters that have at least 16 racks to support all redundancy schemes that we consider (the maximum number of chunks in a coding group is 16, for RS(12,4) and LRC(12,2,2)). To this end, we select 128 clusters from our dataset for evaluation. Due to the varying SSD capacity in our dataset, we fix the capacities of all SSDs as 512 GiB for simplicity. We set the chunk size as 256 MiB, the default chunk size in Facebook [24]. We also fix the same percentage of used storage capacity for data chunks as 50% for each redundancy scheme setting. We set the network link capacity for repair as 1 Gb/s, the parameter used for measuring the repair performance in erasure-coded storage [12, 24]. Furthermore, we set the mission time as ten years and run a sufficient number of iterations for each cluster until the relative error of PDL is less than 20% [31]. As our dataset spans only two years, we replay the dataset from beginning to end repeatedly in each iteration.

5.2 Simulation Results

We first evaluate the reliability of different redundancy schemes based on the SSD failure patterns in our dataset. Figure 18 shows that erasure coding achieves lower PDL and



(a) PDL (b) NOMDL

Figure 18: Finding 13. Comparison of the redundancy schemes.

NOMDL (i.e., higher reliability) than replication. In particular, Rep(2) has the highest PDL (59.9%), indicating that two chunk copies are insufficient to tolerate failures. Also, Rep(3) is not good enough with a PDL of 10.1%. In contrast, RS(10,4) has the lowest PDL and NOMDL among all RS codes, since it tolerates more failures than RS(6,3) and has less repair bandwidth than RS(12,4). LRC(12,2,2) has slightly higher PDL and NOMDL than RS(12,4), since it cannot tolerate four chunks at any time.

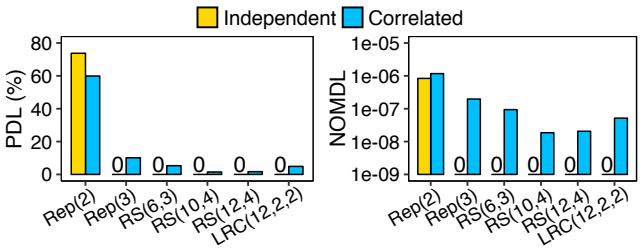
Finding 13. *Erasure coding shows higher reliability than replication based on the failure patterns in our dataset.*

We claim that the redundancy schemes that are sufficient for tolerating independent failures may be insufficient for correlated failures. To justify this claim, we examine the reliability under only independent failures (generated from a mathematical failure model) and under the failure patterns in our dataset (including both independent and correlated ones). Specifically, we generate independent SSD failures following an exponential distribution with the mean time between failures (i.e., the number of hours in a year over the overall AFR in §2.2) in our dataset as the rate parameter, i.e., $\frac{8760}{1.16\%}$.

Figure 19 shows the results of the PDL and NOMDL for eager recovery under only independent failures and the failure patterns in our dataset. The PDL and NOMDL under only independent failures for Rep(3), RS(6,3), RS(10,4), RS(12,4), and LRC(12,2,2) are zero. However, the reliability of these redundancy schemes degrades under the failure patterns in our dataset. The reason is that some correlated failures occur within a short time period (Finding 3) and additional failures are likely to occur in a short time with the existing correlated failures on the same node or rack (Finding 2), leading to the competition for network bandwidth resources and a slowdown of the repair process. This increases the likelihood of data loss. In addition, the PDL under only independent failures for Rep(2) is higher than that under the failures in our dataset by 13.8%. The reason is that the number of failures generated by the mathematical failure model may be more than that in our dataset for some clusters, leading to more failed chunks that exceed the tolerable limit of Rep(2). This implies that Rep(2) is still insufficient under only independent failures.

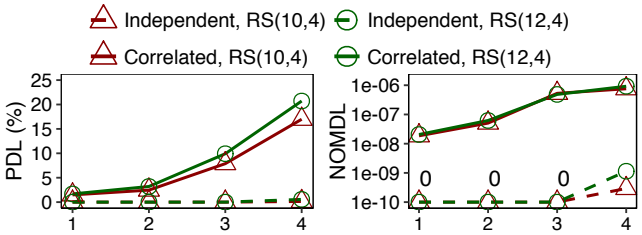
Finding 14. *Redundancy schemes that are sufficient for tolerating independent failures may be insufficient for tolerating the correlated failures as shown in our dataset.*

We next evaluate the reliability of lazy recovery under only



(a) PDL (b) NOMDL

Figure 19: Finding 14. Comparison of the PDL and NOMDL of eager recovery under independent failures ("Independent") and the failure patterns in our dataset ("Correlated").



(a) PDL (b) NOMDL

Figure 20: Finding 15. Comparison of the PDL and NOMDL for the threshold number of chunk failures for recovery under only independent failures ("Independent") and the failure patterns in our dataset ("Correlated").

independent failures derived from the mathematical failure model and under the failure patterns in our dataset. For lazy recovery, we vary the threshold of triggering recovery from one to four failed chunks (note that four is the tolerable limit for RS(10,4) and RS(12,4)); a threshold of one implies eager recovery.

Figure 20 shows that RS(10,4) and RS(12,4) achieve a high reliability under only independent failures, but their reliability degrades under the failure patterns in our dataset as the threshold increases. In particular, under only independent failures, RS(10,4) and RS(12,4) can achieve a high reliability without data loss with a threshold of one to three failed chunks, conforming to the prior work [18]. They have a small PDL (0.14-0.56%) with a threshold of four failed chunks since having any additional failed chunk will lead to data loss. However, under the failure patterns in our dataset, the PDL values for RS(10,4) and RS(12,4) increase by 0.98-1.5% when the threshold increases from one to two failed chunks, and continue to increase by more than 10% from two to four failed chunks. The reason of the reliability degradation of lazy recovery under the failures in our dataset is that when the number of failed chunks reaches a larger threshold of chunk failures, additional correlated failures are also more likely to occur in a short time (Findings 2 and 3). Thus, the most proper threshold number of chunk failures is one, i.e., eager recovery, under the failure patterns in our dataset.

Finding 15. *Lazy recovery is less suitable than eager recovery for tolerating correlated failures in our dataset.*

6 Related Work

SSD measurement. Field studies have analyzed the reliability of SSDs and characterized the correlations between SSD failures and their symptoms [2,15,16,18,27,30]. For example, some studies [16, 18, 27] analyze the symptoms (e.g., uncorrectable errors) reported by proprietary customized attributes and SMART attributes in SSD failures. Xu *et al.* [30] investigate the effects of system-level symptoms on SSD failures. Alter *et al.* [2] exploit the failure patterns from the symptoms to predict future SSD failures. Maneas *et al.* [15] analyze how SSD replacements and other factors affect the replacement rates within a RAID system. Although some studies [15, 16] report the existence of correlated failures in SSD-based storage systems, they do not cover the location details of SSD failures due to the limited information in their datasets. In general, the above studies mainly focus on how SSD failures are correlated with different factors, while our work focuses on the correlations among the SSD failures. In particular, we characterize the correlated failures within a node or a rack. We study the impact of different factors on correlated failures, and the implications on storage reliability under correlated failures in SSD-based data centers.

HDD measurement. Field studies have analyzed the reliability of hard disk drives (HDDs) in production environments. Pinheiro *et al.* [22] analyze different factors that are correlated with HDD failures based on SMART logs at Google. Schroeder *et al.* [26] characterize the HDD replacement rates statistically. Also, prior studies present the patterns of latent sector errors [4, 25] and data corruptions [3] at NetApp. In the literature, Lu *et al.* [14] leverage the locations of HDDs to predict HDD failures. Instead, our work uses the locations to study correlated failures of SSDs.

Correlated failures. Prior studies have characterized the correlated failures on various storage scopes. Chun *et al.* [5] and Nath *et al.* [19] investigate the correlated failures that threaten the durability and availability of storage systems. Schroeder *et al.* [25, 26] provide a statistical analysis on correlated failures of hard disks and the bursts of latent sector errors in disks. Ford *et al.* [8] characterize the statistical behavior of correlated node failures. In contrast, we focus on characterizing the correlated failures in SSD-based data centers in a more comprehensive manner.

7 Conclusion

We present an in-depth analysis on correlated failures of SSDs based on the large-scale dataset at Alibaba. Our analysis includes spatial and temporal correlations of SSD failures and the impact of different factors on correlated failures. We also evaluate the reliability of various redundancy schemes under correlated failures via trace-driven simulation. We report 15 findings, and release our dataset and source code for public validation.

Acknowledgement

We thank our shepherd, Jiri Schindler, and the anonymous reviewers for their comments. We also thank Qiuping Wang and Jinhong Li for their feedback. This work was supported in part by Alibaba Group via the Alibaba Innovation Research (AIR) program and the Research Grants Council of Hong Kong (AoE/P-404/18).

References

[1] R. Alagappan, A. Ganesan, Y. Patel, T. S. Pillai, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. Correlated crash vulnerabilities. In *Proc. of USENIX OSDI*, 2016.

[2] J. Alter, J. Xue, A. Dimnaku, and E. Smirni. SSD failures in the field: Symptoms, causes, and prediction models. In *Proc. of ACM/IEEE SC*, 2019.

[3] L. N. Bairavasundaram, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, G. R. Goodson, and B. Schroeder. An analysis of data corruption in the storage stack. *ACM Trans. on Storage*, 4(3):8, Nov 2008.

[4] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler. An analysis of latent sector errors in disk drives. In *Proc. of ACM SIGMETRICS*, 2007.

[5] B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M. F. Kaashoek, J. Kubiatowicz, and R. T. Morris. Efficient replica maintenance for distributed storage systems. In *Proc. of USENIX NSDI*, 2006.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In *Proc. of ACM SOSP*, 2007.

[7] A. Fikes. Storage architecture and challenges. *Talk at the Google Faculty Summit*, 2010.

[8] D. Ford, F. Labelle, F. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan. Availability in globally distributed storage systems. In *Proc. of USENIX OSDI*, 2010.

[9] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google file system. In *Proc. of ACM SOSP*, 2003.

[10] K. M. Greenan, J. S. Plank, J. J. Wylie, et al. Mean time to meaningless: MTTDL, Markov models, and storage system reliability. In *Proc. of USENIX HotStorage*, 2010.

[11] A. Haeberlen, A. Mislove, and P. Druschel. Glacier: Highly durable, decentralized storage despite massive correlated failures. In *Proc. of USENIX NSDI*, 2005.

[12] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin. Erasure coding in windows azure storage. In *Proc. of USENIX ATC*, 2012.

[13] S. Kadekodi, K. Rashmi, and G. R. Ganger. Cluster storage systems gotta have HeART: improving storage efficiency by exploiting disk-reliability heterogeneity. In *Proc. of USENIX FAST*, 2019.

[14] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, and W. Shi. Making disk failure predictions SMARTer! In *Proc. of USENIX FAST*, 2020.

[15] S. Maneas, K. Mahdaviani, T. Emami, and B. Schroeder. A study of SSD reliability in large scale enterprise storage deployments. In *Proc. of USENIX FAST*, 2020.

[16] J. Meza, Q. Wu, S. Kumar, and O. Mutlu. A large-scale study of flash memory failures in the field. In *Proc. of ACM SIGMETRICS*, 2015.

[17] S. Muralidhar, W. Lloyd, S. Roy, C. Hill, E. Lin, W. Liu, S. Pan, S. Shankar, V. Sivakumar, L. Tang, et al. f4: Facebook's warm BLOB storage system. In *Proc. of USENIX OSDI*, 2014.

[18] I. Narayanan, D. Wang, M. Jeon, B. Sharma, L. Caulfield, A. Sivasubramaniam, B. Cutler, J. Liu, B. Khessib, and K. Vaid. SSD failures in datacenters: What? when? and why? In *Proc. of ACM SYSTOR*, 2016.

[19] S. Nath, H. Yu, P. B. Gibbons, and S. Seshan. Subtleties in tolerating correlated failures in wide-area storage systems. In *Proc. of USENIX NSDI*, 2006.

[20] M. Ovsiannikov, S. Rus, D. Reeves, P. Sutter, S. Rao, and J. Kelly. The Quantcast file system. In *Proc. of the VLDB Endowment*, 2013.

[21] K. Pearson. Vii. note on regression and inheritance in the case of two parents. *Proc. of the Royal Society of London*, 58(347-352):240–242, 1895.

[22] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure Trends in a Large Disk Drive Population. In *Proc. of USENIX FAST*, 2007.

[23] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[24] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur. XORing Elephants: Novel erasure codes for big data. In *Proc. of the VLDB Endowment*, 2013.

[25] B. Schroeder, S. Damouras, and P. Gill. Understanding latent sector errors and how to protect against them. *ACM Trans. on Storage*, 6(3):1–23, 2010.

[26] B. Schroeder and G. A. Gibson. Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you? *ACM Trans. on Storage (TOS)*, 3(3):8–es, 2007.

[27] B. Schroeder, R. Lagisetty, and A. Merchant. Flash reliability in production: The expected and the unexpected. In *Proc. of USENIX FAST*, 2016.

[28] M. Silberstein, L. Ganesh, Y. Wang, L. Alvisi, and M. Dahlin. Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage. In *Proc. of ACM SYSTOR*, 2014.

[29] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471, 1987.

[30] E. Xu, M. Zheng, F. Qin, Y. Xu, and J. Wu. Lessons and actions: What we learned from 10K SSD-related storage system failures. In *Proc. of USENIX ATC*, 2019.

[31] M. Zhang, S. Han, and P. P. Lee. SimEDC: A simulator for the reliability analysis of erasure-coded data centers. *IEEE Trans. on Parallel and Distributed Systems*, 30(12):2836–2848, 2019.