RELIABILITY CHARACTERIZATION AND PERFORMANCE ANALYSIS OF SOLID STATE DRIVES IN DATA CENTERS

Shuwen Liang

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2021

APPROVED:

Song Fu, Major Professor
Yan Huang, Committee Member
Xiaohui Yuan, Committee Member
Hui Zhao, Committee Member
Stephanie Ludi, Interim Chair of the
Department of Computer Science
and Engineering
Hanchen Huang, Dean of the College of
Engineering
Victor Prybutok, Dean of the Toulouse
Graduate School

Liang, Shuwen. *Reliability Characterization and Performance Analysis of Solid State Drives in Data Centers*. Doctor of Philosophy (Computer Science and Engineering), December 2021, 124 pp., 27 tables, 40 figures, 98 numbered references.

NAND flash-based solid state drives (SSDs) have been widely adopted in data centers and high performance computing (HPC) systems due to their better performance compared with hard disk drives. However, little is known about the reliability characteristics of SSDs in production systems. Existing works that study the statistical distributions of SSD failures in the field lack insights into distinct characteristics of SSDs.

In this dissertation, I explore the SSD-specific SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes and conduct in-depth analysis of SSD reliability in a production environment with a focus on the unique error types and health dynamics. QLC SSD delivers better performance in a cost-effective way. I study QLC SSDs in terms of their architecture and performance. In addition, I apply thermal stress tests to QLC SSDs and quantify their performance degradation processes. Various types of big data and machine learning workloads have been executed on SSDs under varying temperatures. The SSD throughput and application performance are analyzed and characterized.

Copyright 2021

by

Shuwen Liang

ACKNOWLEDGMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

First and foremost I would like to thank my advisor Dr. Song Fu for his tremendous help and encouragement during my PhD study. His advising was invaluable for formulating the research questions and developing the methodologies. I was fortunate to have Dr. Fu support my journey of research and connect me with other scientists in collaborations.

I would also like to thank my dissertation committee members, Dr. Yan Huang, Dr. Xiaohui Yuan, and Dr. Hui Zhao for their inspiring ideas and valuable suggestions for my research. They taught me computer systems related knowledge, machine learning related skills, as well as data visualization, which strengthen and broaden my research abilities and vision. I deeply appreciate their comments and suggestions, and making time to attend my proposal exam and dissertation defense.

Last but not least, I cannot have the confidence and achievements without the support of my family. They are always there for me. I would like to thank my husband, my daughter, my parents, my grandparents, and also my dog Lucas from the bottom of my heart for their selfless care, encouragement, and companion. They are my motivation of the PhD journey. Also, I could not complete this dissertation without the support of my lab mates. They provide stimulating discussions on my project and give me a lot of great ideas for development and experiments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND MOTIVATION

Whereas even a few years ago a terabyte was seen as a large amount of data, today individual application can generate petabytes of data per day. Some examples include: 500 terabytes of new data per day are ingested in Facebook databases; launching a series of XGC1 simulations on the Titan supercomputer at Oak Ridge National Laboratory (ORNL) needs to write out 100 petabytes of data in order to capture all of the turbulence data. The explosion of big data continuously pushes the expansion of storage systems in order to accommodate data at this scale.

Over 90% of the world's new information produced annually is stored on magnetic media, most of which are low-cost, high-capacity hard disk drives (HDDs). Datacenter owners all have mission-critical workloads and need to guarantee the quality of service to their customers, which are heavily reliant on their storage systems.

However, disk drives are reported to be the most commonly replaced hardware components, accounting for 78% of all hardware replacements [30]. The annualized failure rate (AFR) of disk drives can reach 15% [75], with 2-4% common for enterprise-class drives and 8-9% for consumer-grade drives. A modern datacenter usually has tens to hundreds of thousands of disk drives installed. At such a scale, disk failures are common with tens of instances every day, not to mention more logical failures that make disk drives inaccessible [52].

The current solutions to disk failures rely mainly on disk rebuilds [70]. As data generated by applications get bigger and so do disk drives, time to rebuild a failed drive is extended, causing tens of hours of data disruption. For helium-filled hard drives, due to the larger capacity and the slower increase in performance, their rebuild time can reach several days. In large storage systems, more disk drives may fail during disk rebuilds, resulting in data loss and significant performance degradation. Storage downtime and data loss cost enterprises $1.7 trillion per year [93].

In recent years, a new trend in storage systems is that NAND flash-based solid state drives (SSD) have been widely adopted in data centers and high performance computing (HPC) systems due to their better performance compared with the traditional hard disk drives. However, little is known about the reliability characteristics of SSDs in production systems. Existing works that study the statistical distributions of SSD failures in the field lack insights to distinguish SSDs from hard disk drives. Moreover, as the SSD technologies advance, the cost-effective quad-level cell (QLC) NAND based SSD has dominated the high-end storage market due to the booming Big Data analytics and data-driven Artificial Intelligence tasks requiring faster data access and larger data storage. As a result, the landscape of data centers evolves rapidly.

In this dissertation research, I investigate SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes for in-depth analysis of the reliability and performance of HDD and SSD in production date center environments. With a focus on the unique error types and health dynamics of SSD architecture, I leverage machine learning technologies, such as data clustering and correlation analysis methods, to discover groups of SSDs that have different health status and relations among SSD-specific SMART attributes.

In the SMART analysis of HDD, I aim to uncover the entire process in which disk's health deteriorates and forecast when disk drives will fail in the future. I model the disk degradation process independent of time, and the intensity of I/O workload, and leverage the derived degradation signatures to forecast when drives will fail in the future. I have developed a prototype of proactive disk failure management system and tested its performance using the SMART data collected from an active data center with approximately 23,000 enterprise-class disk drives, and generalized and verified the methods on a public dataset published by a storage provider.

As the dominant workloads in a modern data center have more read requests than write requests, the latest SSD technologies such as QLC provides a cost-effective solution for data centers' storage systems. In addition to SMART analysis, I study the performance and reliability of QLC SSDs. QLC SSDs deliver better flash storage performance at the cost that is comparable to the traditional hard disk drives and MLC and QLC SSDs. In this dissertation, I analyze the impact of the QLC technology on the landscape of modern data centers. For example, how could read-intensive applications such as real-time analytics and deep learning applications benefit from QLC technologies, while balancing the cost and ensuring data reliability and quality of service? Moreover, I stress test QLC SSDs under different temperatures and different types of workloads. I characterize the performance degradation of QLC SSDs under stress tests.

Besides, I explore the state-of-the-art QLC SSDs from the system architecture's perspective and quantitatively showcase the key advancements against previous technologies. This study includes a comprehensive characterization of all major types of SSDs and discussion of factors that play an important role in affecting SSDs' performance and reliability in real-world environments.

1.1. Total Cost of SSD Reliability

1.1.1. Hardware Cost

Hardware cost, in my studies, is mainly defined as disk retirement and replacement costs. Although the reliability of disk drives has increased over the past few years, disk drives are reported to be the most commonly replaced hardware components in storage systems, accounting for 78% of all hardware replacements[87][76]. Even though some studies showed that the failure rate of SSDs is much lower than that of HDDs [77][10]. A field study on the data collected from Google data centers for over 6 years found that SSDs encountered an infant mortality in their first a few years of use, ranging from 4% - 10% depending on different models. Moreover, as writes and erases to an SSD wear it out gradually, after a certain number of operations, the SSD should retire to keep storage reliable.

1.1.2. Data Management Cost

The booming of cloud computing, edge computing, online services and data-driven artificial intelligence applications results in large scale storage systems. In my studies, the data booming cost refers to the cost that accounts for data processing and data storage due to the rapid increase of the volume of data.

New data generation is tremendous faster than storage developed. In 2014, Forbes sees that every day, human beings create 2.5 Quintillion Bytes(equals to 2.5e+8 Gigabytes.) of data from those sources. Even despite the fact that some data do not go through storage systems, we still can expect more than 150 Zettabytes (equals to 1.5e+14 Gigabytes) of data that will be required to analysis by 2025[21]. Facing such amount of data, storage devices in data centers are facing a great challenge: how to maintain such growing amount of data for a relatively limited storage space and power supply?

1.1.3. Data Recovery Cost

Data Recovery Cost is an additional cost to deal with data integrity, data loss, and data recovery, including time cost as well as other resource costs. The data recovery cost is mass due to some reasons.

First, the storage recovery process is slow and inefficient. In modern data centers, disk drives are assigned to different RAID systems. Disk drives failure leads to the RAID system to process the data restore procedure. The data recovery process is time-consuming. For example, a 4TB drive may take a day or even longer to rebuild. At the same time, when the RAID system involves intensive I/O, its normal performance to applications and operating systems will also be downgraded or even worst, unavailable.

Moreover, the recovery process is passive. The recovery processes of existing storage systems are mostly passive. It occurs only after disk failure happened. In order to rebuild the data, the storage system needs to suffer expensive parity computations before writing down the data to the new disk. Thus, in addition to writing the data to a new disk, computing parities to rebuild the original data takes most of the time and power.

Last but not the least, the data recovery process may lead to more disk failure. The data recovery process is a computation and I/O intensive process. The possibilities of disk failure during the data recovery process is much higher than in normal situations. So, while the RAID system is recovering the data from the first failure disk, there is a second disk that may be about to fail. For RAID 6 and RAID 10, they may tolerate 2 disk failures. But for the most popular RAID 5 that most data centers adopted, 2 disk failure means data is forever lost.

1.2. Motivation

In response to the massive data explosion and thanks to the development of lithography technology, more and more data centers adopt SSD as their new storage devices.Traditional data centers rely on massive arrays of HDD to provide higher aggregated performance, however, switching to SSD storage can achieve paralleled performance with a much smaller data center footprint; this eliminates unnecessary rack space, cooling costs, power consumption and maintenance effort. The memory-storage hierarchy in data centers has been shifting from using HDD as permanent storage to using SSD or a fusion of Flash cache and HDD storage. While their deployment is increasing, the endurance of SSD still remains as one of the main concerns.

Ever since SSD were introduced to the enterprise market many years ago, its density continues to improve thanks to the advances of semi-conductor technology. Depending on the number of bits stored in each flash cell, there are four basic types of NAND flash used in an SSD. Each type have its own distinct performance, cost, endurance, and density trade-offs. Single-level cell (SLC) requires 2 voltage levels (i.e., 0 and 1) to store 1 bit of data, offering highest write performance and endurance at the cost of price and density. Multilevel cell (MLC) requires 4 voltage levels to represent 2 bits of data (i.e., 00, 01, 10, and 11). Triple-level cell (TLC) and Quadruple-level cell (QLC) requires 8 and 16 levels of voltage to store 3 and 4 bits of data, respectively. As the data density increase, the price per bit lowers, but the endurance and write performance decreases as a result.

Because endurance is such a vital component to the SSD, it is the primary concern to many data centers that are adopting SSDs. As SSD technology uses NAND flash, the inherent wear-out characteristics of the NAND flash directly effects the durability of the SSD. NAND flash chips are non-volatile, meaning they retain data without a constant power supply. Furthermore, because NAND flash based SSDs does not have moving parts involved during operation, they are more resistant to sudden shocks and extreme environments than their HDD counterparts. On the flip side, the NAND flash will eventually wear out as data writes accumulate overtime. If the amount of data written to the device exceeds its life span, NAND flash based SSD will gradually lose the ability to retain charge and the ability to retain data integrity. Even though SSD employs several techniques to improve endurance, e.g., wear leveling, garbage collection, and chip-level RAID, performance and reliability degradation is irreversible. A deeper understanding of SSD endurance and reliability in data centers is vital.

1.3. Organization of this Dissertation

In first part of this research, I perform an in-depth analysis of SSD reliability using SSDs' SMART data collected from an scalable, active production environment. The SSDs are used as caching devices from approximate 300 servers spread across several data centers in the United States. The SSDs in my dataset are all MLC SSDs that considered the most reliable flash type than others. The dataset contains six-months of SSD SMART data. I use machine learning models and statistical analysis to investigate more than 20 SSD-specific performance and error-related SMART attributes from over a million complete records.

In the second part of this research, I evaluate the QLC technology's impact on the landscape of modern data centers. I study the latest QLC technology, which improves the SSD economics and fills the much-needed gap between MLC/TLC SSD and legacy HDD storage, from an architecture level to evince its key advancements over previous technologies. This storage paradigm transition enables more read-focused workloads to be migrated from dated HDD based storage to flash, thus releasing the expensive and limited MLC/TLC resources for write-centric applications. I evaluate two real-world QLC SSDs performance and compare them against state-of-art SSDs using MLC and TLC technology.

The third part of this research focus on the stress test of QLC SSD. I discuss the increasing temperature effecting on the performance degradation. In this section of the dissertation, performance degradation is quantified and evaluated under different temperatures and different types of workloads.

In the following part, I develop a proactive disk failure management system for HDD. The disk degradation model and proactive management method can capture the dynamics of disk health and accurately forecast failure occurrence time, which enables datacenter operators to protect the system and user data before the disk failures really happen.

In the last part of the dissertation, I develop an on-drive reliability management paradigm – ACTOR, using open Ethernet drives to address the scalability and reliability of data. I test the the ACTOR and evaluate performance of the platform.

The remainder of the dissertation is organized as follows. Chapter 2 presents the background knowledge of the disk drives, storage system reliability, and the architecture of 3D NAND. Chapter 3 describes the data sets and analysis methods. Chapter 4 characterizes SSD endurance and reliability analysis. In the next chapter, I evaluate the QLC SSD performance for modern data centers. Chapter 6 discusses the QLC SSD performance in acceleration tests. And chapter 7 discusses the failure prediction prototype for HDD. Chapter 8 introduces the data placement design ACTOR via open Ethernet drives and evaluates its performance. The last chapter concludes the paper with remarks on future research.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1. Fault Modes of Disk Drives

Most disk drives do not fail in a simple fail-stop way. The production data center that I study defines a disk failure in three cases: the system loses connection to the disk, an operation exceeds the timeout threshold, or a write operation fails. Those drives that cannot function properly are replaced.

Operations to drives may be invoked by read and write calls from the file system as well as by an internal disk scan process which checks sectors' reliability and accessibility in the background. There are several types of disk errors. Read or media error occurs when a sector cannot be read either during a normal read operation or in a background disk scan. Data previously stored in the sector is lost. A status code, specifying the reason that the read fails, is reported. Reallocated sector happens after a number of unsuccessful write retries. The drive re-maps a failed write to a spare sector. Disk drives usually reserve several thousand spare sectors. Unstable sectors detected in the disk scan process are marked as pending sectors. Disk drives then try to correct the errors using Error Correcting Code (ECC). Sectors that cannot be successfully recovered are called uncorrectable sectors. Seek errors occurs when a disk drive fails to properly locate a track and needs another revolution to read from or write to a sector.

Similar to HDD, SSD manufacturers have their proprietary FTL policies to manage ECC, wear-leveling, and garbage collection. Thus there exists a variety of reliability characteristics of different SSDs. For example, reading and writing data causes wear of flash cells, which degrades SSD reliability gradually. A number of prior works studied the correlation between wear and increased of error rate[37, 41, 62, 63, 65, 77]. Wear-leveling is designed to distribute data across SSD to address this issue.

Retention errors that are caused by the leakage current increase with usage [29, 38, 34, 44, 56, 94]. If not confined or corrected in time, retention errors quickly propagate. As read and programming operations can affect the threshold voltage of the neighboring blocks[31, 33, 53, 54, 85], causing untouched cell susceptible to *read disturb errors* and *program disturb errors*. Besides wear-leveling, FTL uses ECC as a countermeasure to prevent the above-mentioned error propagation to the upper data hierarchy, that is checksum in each page of the spare space is used by ECC to protect against these errors. If the number of bit errors exceeds the capability of page-level ECC, SSD controller performs error correction at the host driver level by using more complex error correction algorithms. From the SSD's perspective, page-level ECC are correctable errors, while host driver level ECC are uncorrectable errors. When the number of uncorrectable errors in a block exceeds a preset threshold, FTL marks it as a bad block and remaps the data to a reallocated block in the spare space. SSD manufacturers are conservative about the *endurance rating* and reserve a considerable amount of space for remapping data from bad blocks.

2.2. Anatomy of SSD Architecture

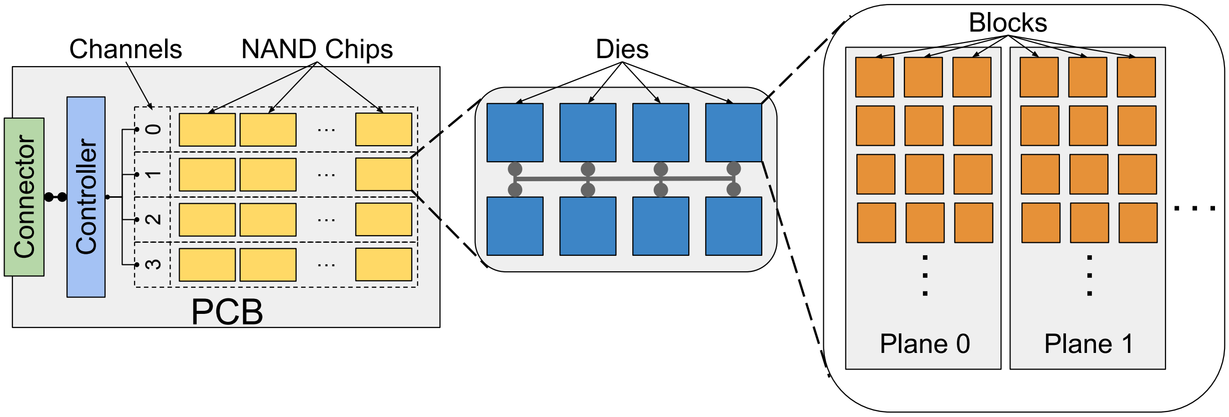


FIGURE 2.1. Example of an SSD Physical Layout.

In this section, I discusses the general architecture of SSD. From physical to logical, this section mainly focuses on the construction of the SSD and how data flows in each SSD components.

Physically, an SSD mainly consists of the following components.

- Controller.

- NAND flash memory chips.
- Connector.
- Integrated Circuits (ICs).
- Resistors, inductors, and capacitors.
- Printed Circuit Board (PCB)

2.2.1. Controller

Table 2.1. Features of SSD Controller

Feature*	**Description**
Flash Translation Layer (FTL)	Map logical address (LBA) to physical address in the flash memory [59].
Bad Block Mapping	Map the bad block's logical sector to reserved sector when bad block is detected [18].
Wear Leveling	The mechanism that maps re-writed/updated data to a new location and marks the previous location as "invalid". Meanwhile, cold data will also be moved around periodically to provide evenly wear among each cell [36].
Error Correction (ECC)	Detect and correct memory bit errors(or soft errors). Hamming code and parity bit error detection schema are widely used in SSD [18].
S.M.A.R.T.	Monitor and report SSD health status. Some SSD S.M.A.R.T. attributes are pre-defined, some are manufacturer defined [17].
Encryption/Decryption	Controller support encryption/decryption to ensure data security; it typically uses the 256 AES encryption. Encryption can be applied to partial or whole drive.[18].

Garbage Collection	the controller erase invalid data blocks periodically to ensure the available space for new write request. Garbage collection typically runs in the background during idle time [18].
I/O Caching	store frequently used or recently used data to exploit spatial and temporal locality.
Data scrubbing	The mechanism that verifies the integrity of each memory block periodically. If a bit error is detected, controller will invoke ECC to correct in the same memory location. Data scrubbing is usually operated in disk idle time [18].
Power Management	To improve power efficiency for SSD, especially when used in mobile devices, the controller manages power consumption for the SSD in different states: active, idle, and slumber.
Trim Support	Enables the operating system to notify SSD on which data blocks can be erased [20].
Thermal Throttling	With the internal thermal sensors monitoring the environment temperature, SSD controller will reducing the I/O speed when overheating.

*Features of proprietary controllers are not included.

An SSD controller works like a central manager that in-charge of all the NAND chips in the drives. The modern SSD controller is also a powerful ”brain” as it is capable of managing different kinds of jobs; it executes firmware-level code, manages I/O requests, and ensures data integrity and storage efficiency. In particular, it manages bad blocks, enforce wearing leveling, monitors disk health, and handles garbage collection. Table 2.1 summarizes universal features that are supported by most SSD controllers. Each SSD manufacturer also has its own unique tweaks or features to the SSD controllers that boost the performance and reliability.

2.2.2. NAND Flash Memory Chip

NAND chips are important components to the SSD. Thus far, an SSD can have 4-16 NAND chips [59]. An individual NAND chip can be decomposed from top to down in the following order (Also see Figure 2.1):

- **Die** : Each NAND chip can contain several NAND memory dies.
- **Plane**: Each die can contain 1-4 planes [59].
- **Block** : Each plane has thousands flash blocks.
  - Page/Wordline[1] : Each block contains hundreds to thousands of rows of pages (horizontal).
  - String/Bitline[2] : Each block contains hundreds to thousands of columns of strings (vertical).
- **Cell** : Each page or string contains thousands of flash cells.

In 2D NAND, A flash block is the cluster of wordlines and bitlines. Figure 2.2 shows the details on the architecture of a block. A row is called a wordline, and a column is called a bitline. In the block level, transistors are neatly arranged.

The *Page* is the smallest data storage unit that can be read and wrote to, while the *Block* is the smallest data storage unit that can be erased. A page can typically contain 2K, 4K, 8K or 16KB data. And, the size of a block can vary between 256KB and 4MB [59]. But as technology develops rapidly, we can expect these numbers update quickly.

[1] When talking about data, we usually use ”page”; while referring to architecture, we use ”wordline” more often.

[2] When talking about data, we usually use ”string”; while referring to architecture, we use ”bitline” more often.

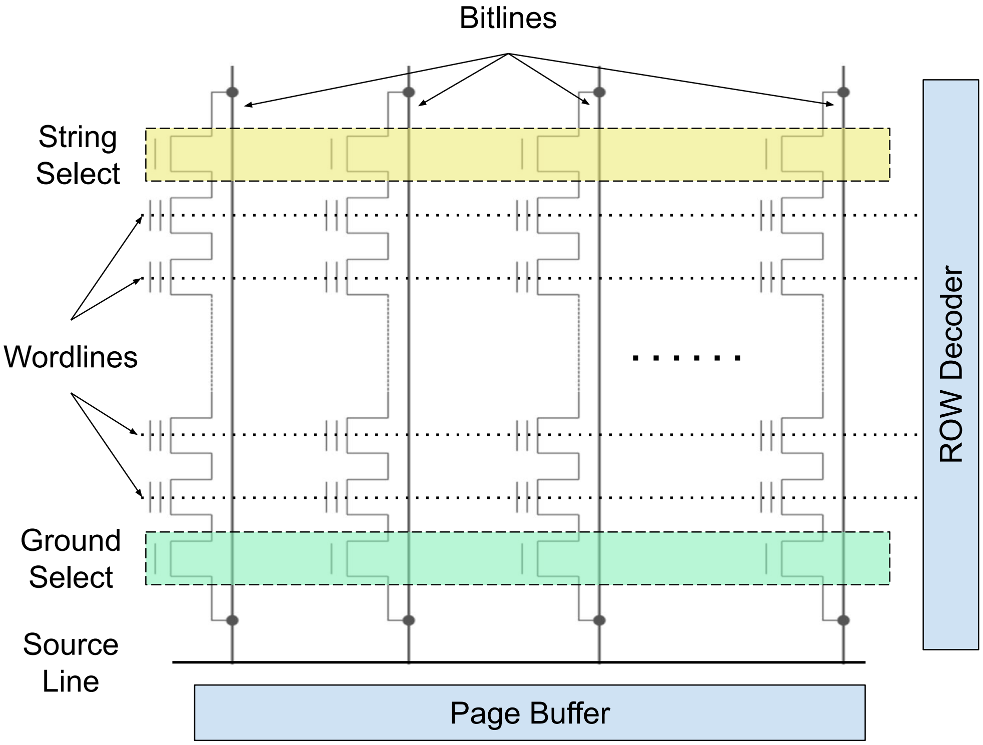


FIGURE 2.2. Example of SSD Block Layout.

Moreover, the number of bits a cell stores defines the performance expectation of an SSD. The cell can only store one bit called Single-Level-Cell(SLC), store two bits called Multiple-Level-Cell(MLC), the same definition method can also apply to Triple-Level-Cell(TLC) and Quad-Level-Cell(QLC). Thus, block and cell are critical units related to data in SSD.

2.2.3. Connector

The connector mediates data and signals transfer between the SSD and the host computer. The universal connectors interface are SATA and PCIe. SATA has three major revisions. Most of the consumer-graded SSD supports SATA 3.0 or above. SATA 3.0 can support a burst throughput up to 6 Gbit/s [16]. In contrast, enterprise-graded and high-end SSD usually embeds the PCIe connector. It applies the NVMe protocol to access NAND flash memory via a PCIe bus. Theoretically, the throughput of PCIe based SSD can be up to 32 Gbit/s [15]. Besides, more and more M.2 and U.2 connectors apply to the SSD drives in recent years. They are designed to connect via the PCIe bus as well. Where, M.2 can be

SATA 3.0 interface or PCIe 3.0 × 4 interface. U.2 is PCIe 3.0 × 4 interface only.

2.2.4. PCB, ICs and Other Components

All components, including controller, NAND chips, and connector, are on the printed circuit board (PCB), connecting with integrated circuits (ICs). Besides, there are sensors and counters tied to the PCB such as the thermal sensor and physical event counters. All components work together to make the SSD work appropriately as a whole.

2.2.5. Data Operations on SSDs

Writes. Data from the file system goes through the SSD connector before arriving at the controller. Since data can only be written to empty blocks, the controller maintains a pool of empty blocks. If the drive runs out of empty blocks, the controller will perform garbage collection to reclaim ”invalid” data blocks before writing data to the NAND memory. Otherwise, the FTL runs the address mapping algorithm and determines the physical addresses in NAND chips – the FTL maps the logical data blocks into the NAND page and then written into a block. Specifically, the controller applies a high positive voltage to responsive NAND pages and strings. Voltages of selected cells will be changed to logical “0“. After related cell voltages are updated, the ECC will verify the written data before return the ”success” signal to the OS.

Reads. Reading data from SSD is similar to the writing process. The file system issues the read request. The request goes through the connector and enters the SSD controller. The controller processes the request and communicates with the NAND interface. FTL locates the physical addresses of the request data. Then, the controller applies the read voltage (intermediate positive voltage) to related NAND pages and strings. Since medium positive voltage won’t change the logical representation of NAND cells, the selected NAND cells will respond with the corresponding stored logical 0 and 1. Raw data then goes through the NAND decoders. After decoding and verifying, data stream sends back to the file system.

Erase. Erasing data in SSD is quite different from erasing data on a HDD. In SSD, the erase process can only be performed in block-level while writing and reading process can perform at page-level. An erase operation is the process of removing electrons from the storage layer to change the state of the cell to logical 1. Typically, delete request sent from the file system won't immediately remove the data from flash memory; the controller only marks the "erased data" as "invalid." The garbage collection algorithm run in the background decides when to issue a large negative voltage to erase the whole block.

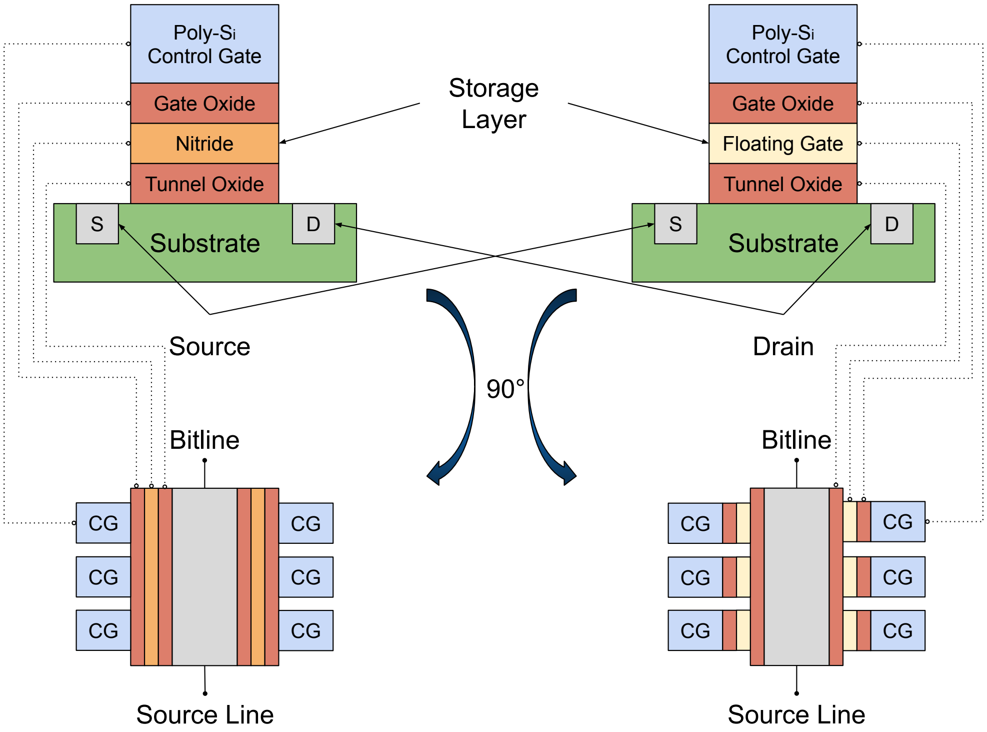


FIGURE 2.3. Charge Trap (CT) cell vs. Floating Gate (FG) cell.

2.2.6. Data Placement

In general, SSD has two storage areas: main area and spare area. Main area stores user data and the spare area contains bad block marker, ECC and may have some metadata. Usually, the spare area is reserved and user cannot get access to it.

Data in SSD, including data placement, are specified and managed by the controller. SSD only write to one page each time and block marked as "bad block" will not be used. But, determining which page will be written and how to skip the bad block are defined by the related algorithms embedded in the SSD controller. The controller also defines other jobs related to data placement. For example, wear-leveling algorithms and I/O algorithms in the controller will divide large files and store each part in different flash chips. The data integrity algorithm writes data parities to a separate flash chip.

2.3. The State-of-the-Art QLC Architectures

The quad-level cell(QLC) technology was first introduced by NEC in 1996 [14]. Compared with prior generation, i.e., SLC, MLC, and TLC, QLC technology enables larger per bit capacity while lowering the cost. This technology was initially developed for the dynamic random-access memory(DRAM) chip. In the 2000s, QLC technology applied to NOR flash memory cells and NAND flash chips. Later on in the winter of 2018, the first commercial QLC NAND-based SSD became available [14]. In this section, I present the key advancements of QLC technology that uplift the storage density, and bridge the gap of performance and cost between flash and legacy HDD storage. The main difference between QLC and other types of SSD lies at the cell level and block level. I highlight the main features of QLC technology in Table 2.2, then explore them in detail and compare them against competitive SSD technologies.

2.3.1. Cell-Level Architecture

2.3.1.1. Physical Architecture

NAND-based SSD generally employs one of the two transistor technologies: Charge Trap (CT) MOSFET (Metal Oxide Semiconductor Field Effect Transistor) and Floating Gate (FG) MOSFET. Manufactures like Samsung, Toshiba, SanDisk, and Western Digital develop their SSD architecture using the CT MOSFET, while Micron and Intel adopt the FG MOSFET. Figure 2.3 illustrates the difference between CT cell and FG cell. The storage layer of CT cell uses the silicon nitride while the FG cell uses floating gate [79]. Additionally, the charged storage layer in CT is shared among all cells while in FG they are isolated. According to Micheloni's study [73], the majority of SSD relying on CT cells but FG cell also have its market share.

Table 2.2. Specifications of SLC, MLC, TLC, and QLC

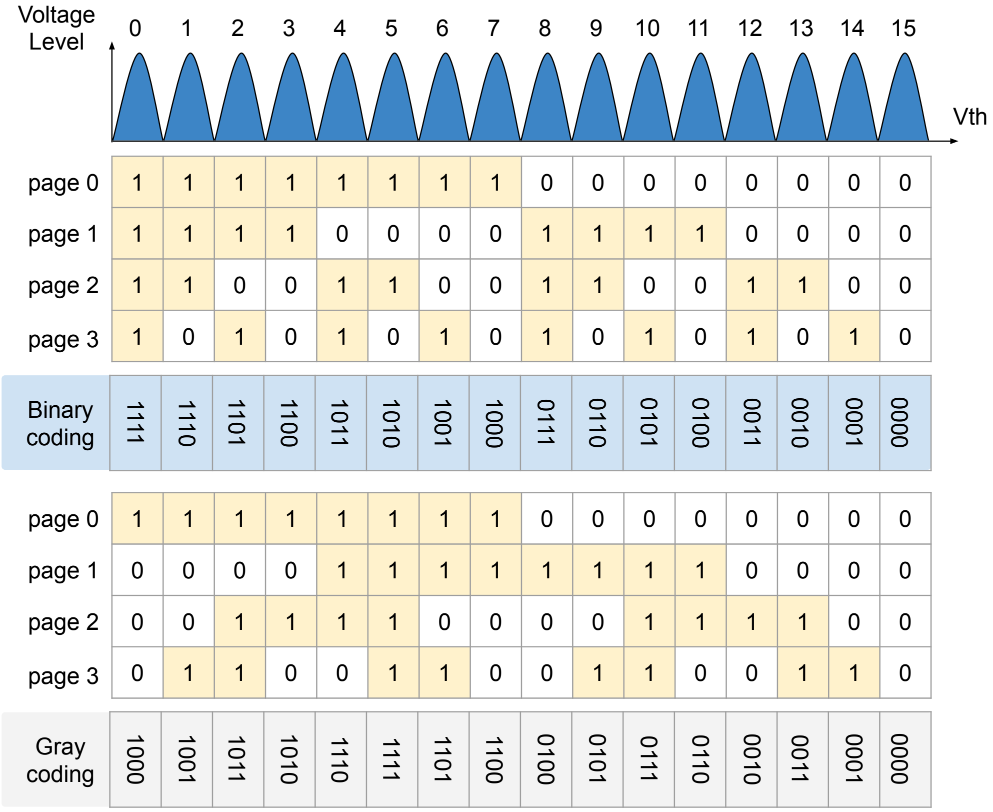
Types	**SLC**	**MLC**	**TLC**	**QLC**
P/E Cycle	90-1000k	8-30k	3-5k	500-1k
Bit per Cell	1	2	3	4
Reliability	*****	****	***	**
Endurance	*****	****	***	**
Power*	0.1-3.6W	0.6-2.6W	0.7-3.6W	1.5-3.6W
Cell Density	*	**	***	****
Voltage Levels	2	4	8	16
NAND Architecture	2D/3D	2D/3D	3D	3D
		Latency / QoS[90]		
Read	25μs	50μs	75 μs	≈100μs
Write	200-300μs	600-900μs	900-1350μs	≈1500μs
Erase	1.5-2ms	3ms	5ms	≈6ms

*Ranging from *idle* to *active* power consumption.

2.3.1.2. Data Programming

One QLC cell can store four bits of information, which is 33% more than TLC technology. As illustrated in Figure 2.4, it requires 2^4 different electric voltage levels to program a QLC cell (represented by 0000 - 1111). A logical QLC cell data (4 bits) are mapped to four pages. Thus, it takes four cycles to raise the voltage level to the desired state.

There are many QLC programming algorithms. Traditional method named Binary Code is illustrated in the upper portion of Figure 2.4. For a narrower and tighter voltage range, traditional data mapping based on Binary Code becomes inefficient and can easy introduce data error [58]. Thus, more sophisticated programming algorithms have been proposed. Since QLC SSD is designed for read-intensive workloads, data reading based on Gray Code method provides a better solution (refer to lower portion of Figure 2.4). The most obvious benefit to this is that two successive values differs in only one bit [13]. Thus, if voltage shift and data retention error occurs, Binary Coding may encounter up to 4 bits data error (i.e., voltage level 7 to 8 in Figure 2.4), Gray coding only yields 1-bit data error. There are many Gray Code variants [58], but all of them retain the 1-bit differ rule for sibling values.



page 0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
page 1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
page 2	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
page 3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Binary coding	1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000

page 0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
page 1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
page 2	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
page 3	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
Gray coding	1000	1001	1011	1010	1110	1111	1101	1100	0100	0101	0111	0110	0010	0011	0001	0000

FIGURE 2.4. QLC Voltage Levels and Data Mapping under Binary Coding vs. Gray Coding.

2.3.2. Block-Level Architecture

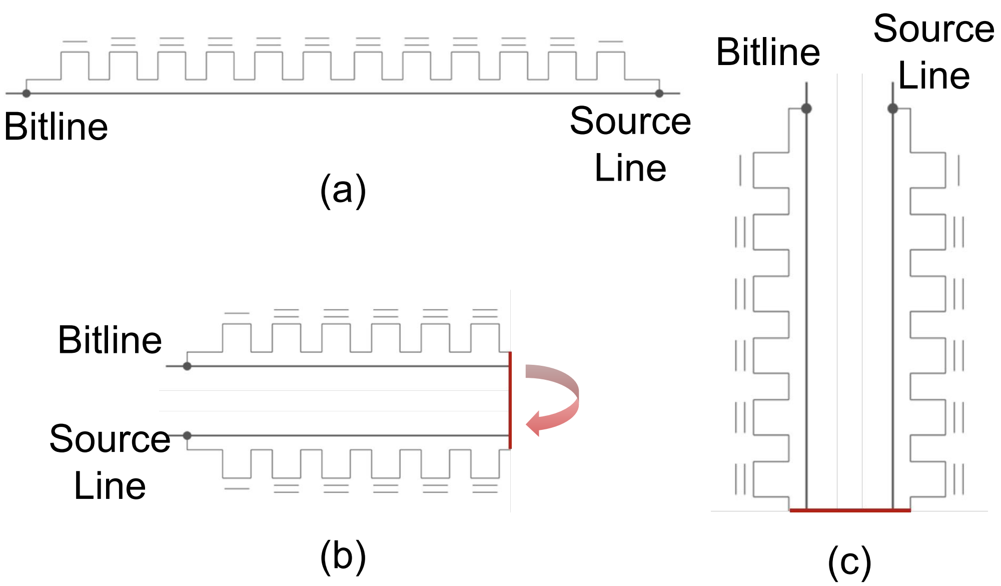
Starting from the TLC technology, block-level design shifted from 2 to 3 dimensional. Each generation of 2D NAND architecture shrinks the size and increases the number of transistors in the chips to accommodate the Moore Law. However, 2D NAND design has reached the lithographic limitation [73]. The new 3D flash architecture extends the vertical space to achieve a higher density and capacity. Thus, QLC SSD continues the trend of using 3D NAND architecture in block level design.

Each manufactures has its proprietary 3D NAND designs. Toshiba developed a 3D NAND technology named Bit-Cost Scalable NAND technology (BiCS) then later improved to Pipe-shaped Bit Cost Scalable (P-BiCS). Samsung introduced several vertical gate (with either horizontal or vertical channels) designs, namely V-NAND architecture. The V-NAND family includes the Vertical Recess Array Transistor (VRAT), the Vertical Stacked Array Transistor (VSAT), and the Terabit Cell Array Transistor (TCAT) [73]. SK Hynix proposed designs based on FG cell, named Dual Control-gate with Surrounding Floating-gate (DC-SF) and its variant called Advanced DC-SF [89].

3D NAND design is an extension of 2D NAND that adds the vertical dimension. Figure 2.5 illustrates a typical way to convert a 2D string to a 3D string. Imagine when all the strings in a 2D block are folded over then stood vertically, essentially transitioning the 2D planar block into a 3D block. Samsung's TCAT and SK Hynix's DC-SF uses different approaches. In a nut shell, TCAT and DC-SF don't fold over the string, but add a "z" axle to the 2D planar instead. Figure 2.6 shows this type of 3D NAND block construction. Both CT and FG cells presented in Figure 2.3 can be applied to this type of 3D architecture.

Transitioning from 2D to 3D block, a new concept called "layer." The number of layers defined by the number of vertical Control Gates (CG). In QLC, the number of layers are usually the multiple of 4. The common construction of 3D NAND flash uses 32, 36, 48, 56 and 96 layers. More layers usually means higher storage density. In the year of 2019, 96 layer 3D NAND dominates the QLC market share. Meanwhile, the 3D NAND flash of over 100 layers (e.g., 128 layers) is just around the corner.

Different 3D NAND block poses various design challenge. I can characterize them from the following aspects: (1) cell size (in μm^2) and shape (gate-all-round or planar) ; (2) cell storage type (CT or FG); (3) channel alignment (horizontal or vertical); (4)gate process preference (gate-first or gate-last). 3D NAND design is sophisticated and thus, 3D NAND architecture remains a research topic.



(a) The 2D NAND string (b) 2D string is stretched out in the middle and folded over (c) Make it stand vertical and it becomes a 3D string.

FIGURE 2.5. P-BiCS Converts a 2D NAND String to 3D NAND String.

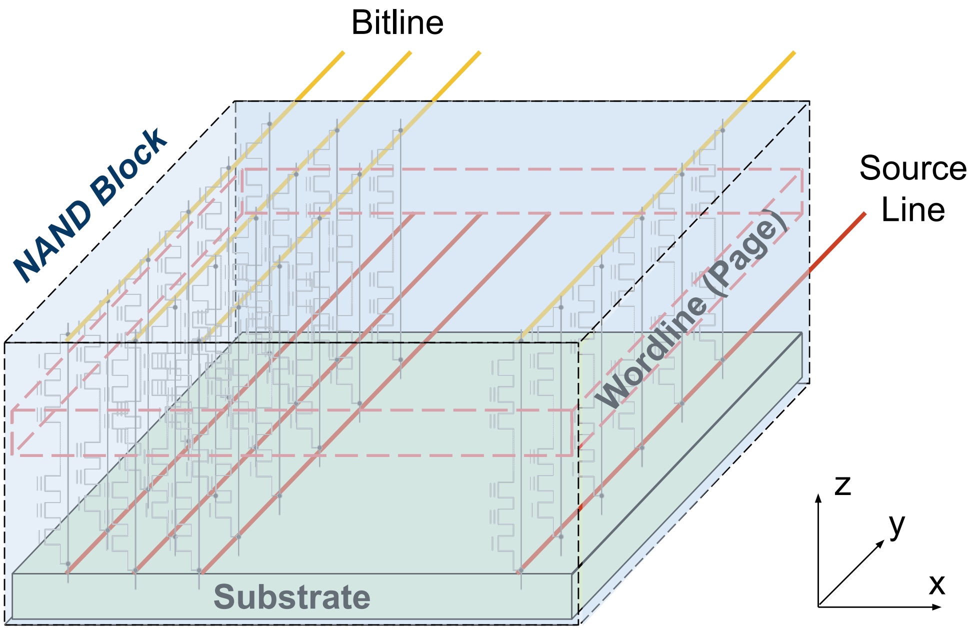


FIGURE 2.6. TCAT and DC-SF Construct a 3D NAND Block.

CHAPTER 3

DATA AND METHODS

3.1. Introduction of Datasets

This chapter discusses the 2 major datasets I used for this research. The first data set is the SSD SMART datset. The SSD SMART dataset was collected from several active production data centers owned by a major financial service provider. Each server in the data center has 2 sockets, 10-18 cores per socket, 1.5 TB of DRAM, 1-2 SSDs, and 8-45 HDDs. There are two types of SSD in my dataset and each type of SSDs have the same age. SSDs are used as a data buffer between the main memory and the storage subsystem. They accelerate data accesses, improve I/O throughput, and reduce access latency from main disks in a Ceph storage node. The workloads run on the servers include investment/portfolio management, brokerage order management systems, and financial planning management. This represents a wide range of operating system main disks, persistent database storage, and sequential message queues.

The SMART monitoring system is employed for both HDDs and SSDs to collect access information and fault/error indicators. SSD SMART data are collected hourly at runtime. By the time of this study, I have six months of SMART records from both HDDs and SSDs. I pre-process the raw SMART data by extracting SMART attributes and their values and removing incomplete records. Over a million SSD records are kept with over 20 attributes from two SSD brands and models. Then, I analyze the correlation among these attributes and further explore machine learning technologies to characterize and study SSD reliability.

Another dataset is the HDD SMART dataset for the failure prediction research in the dissertation. The HDD SMART data used in my discussion and illustration was collected from an active, production data center, equipped with over 23,000 enterprise-class hard disk drives and running Google-like workload, including search, news, multimedia (music, image and video playing), transactions, maps and etc. The disks used in this storage system are from the same manufacturer and the dataset covers a running period of eight weeks. Every hour, the SMART attributes are profiled from every disk drive.

The drives that were replaced are labeled as “failed” drives. Those drives that still function in the storage system are labeled as “good” drives. Overall, more than 400 drives are “failed” in the eight-week period and about 23,000 drives are “good” in the dataset.

Also, I do another set of experiments on a public dataset provided by Backblaze, which provides B2 cloud storage and backup services. Backblaze has published 5-year SMART data collected from 125,000 hard disk drives in its storage system. Those drives come from five manufactures with multiple models. I select the one that has the largest population (i.e., Seagate drives with model ST4000DM000). The dataset contain SMART records from 36,924 drives. The drives that are replaced are labeled as “failed” drives. Those drives that still function in the storage system are labeled as “good” drives. Among the 36,924 drives, 2,811 are failed.

3.2. SMART

S.M.A.R.T. is short for Self-Monitoring, Analysis and Reporting Technology(SMART). It is a monitoring system included in computer hard disk drives HDDs and SSDs. The disk attributes, collected by SMART, directly or indirectly indicate hard drive health status and others give statistical information. Normally, SMART attributes are selected by manufacturers and come with disks. User cannot define and modify which attributes would like to collect.

Each SMART attribute has Identifier : the meaning of the attribute; Data: raw measured values provided by a sensor or a counter; Threshold: the failure limit value for the attribute; Value: the relative health of the attribute (This value is calculated by algorithms, usually linear functions, designed by drive manufacturers, using the raw data. The value is initially set to a theoretical maximum and decreases during the lifetime of the disk); Status flags: indicating the main purpose of the attribute, e.g., critical or statistical (does not directly affect disk condition). Thresholds are pre-defined for each attribute by the drive manufacturer. When the value of an attribute is below its threshold, a warning flag is issued.

3.2.1. SSD Specific SMART Attributes

The SMART technology monitors drives' accesses and errors, and provides various attributes, many of which are particularly designed for SSDs. An SSD manufacturer adopts a subset of all SMART attributes which may be different from that of another manufacturer. Even among drives produced by the same manufacturer, those of different models may have different sets of SMART attributes. In general, I group the SMART attributes into three categories, i.e., environmental factors (e.g., temperature and power-on hours), workload-related statistics (e.g., the amount of data read from or written to flash chips), and error attributes (e.g., the number of seek errors and the number of uncorrectable errors). In the data center that I study, two SSD models from different vendors are found. Table 3.1 lists the SMART attributes provided by the SSDs.

Table 3.1. SSD SMART Attributes and Description*

ID	**Attributes**	**Descriptions**
Environmental Attributes		
9	POH	Power On Hour
12	PCC	Power Cycle Count
174	UPLC	Unexpected Power Lost
194	TC	Temperature Celsius
Workload-related Attributes		
166	MWEC	Min Write/Erase Count
168	MEC	Max Erase Count
173	AWEC	Average Write/Erase Count
180	URNB	Unused Reserve NAND Blocks
202	PLR	Percent Lifetime Remaining
230	PWEC	% of Write/Erase Count
232	PARS	% Avaliable Reserved Space

Continued on next page

Table 3.1 – SSD SMART attributes and descriptions

ID	Attributes	Descriptions
233	TNWG	Total NAND Write(GB)
241	TWG	Total Write(GB)
242	TRG	Total Read(GB)
246	CHSW	Cumulative Host Sectors Written
247	NONPWYTH	Number of NAND page written by Host
248	NONPWNTF	Number of NAND page written by FTL
Error-related Attributes		
4	RRER	Raw Read Error Rate
5	RSC	Reallocated Sector Count
167	MBB	Min Bad Block/Die
169	TBB	Total Bad Block
171	PFC	Program Fail Count
172	EFC	Erase Fail Count
183	SID	SATA Interface Downshift
184	ECC	Error Correction Count
187	RU	Reported Uncorrected
196	REC	Reallocation Event Count
197	CPEC	Current Pending ECC Count
206	WER	Write Error Rate
212	SPE	SATA Physical Error

* Note: Attributes exclusive to Model-A SSDs are colored in *red* and attributes exclusive to Model-B SSDs are colored in *blue*. The rest of the attributes are common for both SSD models.

3.2.2. HDD Specific SMART Attributes

SMART attributes for HDD have some differences from the SDD ones. Due to the different physical structures of the HDD, some SMART attributes are only available to HDD. SUT, SER, and HFW from the Table 3.2 are examples of SMART attributes that are only applicable to HDD. However, some attributes are shared within SDD and HDD, especially some environment-related attributes. Similar to the SSD SMART attributes, disk manufacturers define what attributes are provided to users. Table 3.2 shows the SMART attributes that collected from the HDD to evaluate reliability.

Table 3.2. HDD SMART Attributes

Attributes	**Descriptions**
Spin-Up Time (SUT)	Average time of spindle to spin up to full operation
Seek Error Rate (SER)	Rate of seek errors of the magnetic heads
Power on Hour (POH)	Count of hours in power-on state
Reported Uncorrectable Error (RUE)	Errors that cannot be recovered using ECC
High Fly Writes (HFW)	Errors that head fly outside normal operat- ing range
Hardware ECC Recovered (HER)	Time between ECC-corrected errors
Temperature Celsius (TC)	Current internal temperature in Celsius
Read Error Rate (RER)	Rate of hardware read errors when reading data
Reallocated Sectors Count (RSC)	Count of reallocated sectors when finds an error
Current Pending Sector Count (CPSC)	Current count of unstable sectors (waiting for remapping)

3.3. Evaluate Reliability and Performance

3.3.1. Correlation Analysis

Compared with several decades of deployment of HDDs in the field, SSDs are still at the early stage of usage as the mainstream storage media in production systems. Little is known about SSDs' reliability characteristics in real-world settings. Most recently, large-scale field studies, such as [77], identify substantial differences from those SSD fault data collected in controlled environments. Production systems involve a wide range of conditions, e.g., real-world applications display a variety of access patterns and frequencies compared with synthetic benchmarks. Additionally, the entire software stack on top of a flash storage also affects data accesses, SSD performance degradation, and lifespan.

In the Chapter 4, I investigate the relationship among SMART attributes by quantifying pair-wise correlation coefficients. I also use *boxplot* to visualize the distributions of SSD SMART data. By analyzing the correlation among SSD SMART attributes, I obtain a better understanding of the influence among various factors and their criticalness for characterizing SSD reliability. I compare several correlation coefficients, i.e., Pearson, Spearman, and Kendall. I select the Spearman rank correlation coefficient, because it provides the best modeling of monotonic linear/non-linear relations which are common for SMART attributes. To prevent invalid correlation, I remove SMART attributes whose values remain constant over time.

3.3.2. SSD Evaluation of Reliability and Performance

In order to understand SSD's reliability and discover the relations between SMART attributes and SSD's health status at the drive level, I analyze the SSD SMART dataset by using machine learning methods. Specifically, I explore K-means clustering on SMART records. K-means is an unsupervised machine learning method, which is used to discover groups of data items with similar feature patterns. It can help us establish a dynamic view of SSD's health status with possible transitions over time.

3.3.3. Evaluating Performance and Reliability of QLC SSD

This dataset is collected by my experiments. It includes SSD performance as well as SMART data. Historically, performance-sensitive and read-centric workloads have relied on parallel arrays of HDDs to deliver the required capabilities that service-level agreement (SLA) demand. With the advance of QLC technology, can these new SSD achieve the storage performance and capacity requirement? I evaluate the real-world performance of the latest QLC SSDs and compare its performance with state-of-art SSDs using MLC and TLC technologies. I try to answer the question: *Can QLC SSD offers flash storage performance* *at more approachable price?* In this project, I will address these problems.

3.3.4. Acceleration Tests of QLC SSD

In the acceleration tests of QLC SSD, the dataset is collected by in-lab experiments. Each test is repeated 100 times to allow the disks to reach an stable status. I collect data from each time of experiment running. The dataset includes disk performance data, workload data and temperature data. The sources of the data come from different monitoring tools and devices. I mainly apply statistical theologies to analyze the data, for example, performing T-test to evaluate the significant level of the performance degradation. The P-value is calculated by the following formula:

$$P - value = \frac{mean1 - mean2}{\frac{(n1-1)\times var1^2+(n2-1)\times var2^2}{n1+n2-2} \times \sqrt{\frac{1}{n1}+\frac{1}{n2}}}$$

where,

mean1 and *mean2* are average values of each sample sets,

var1 and *var2* are variances,

n1 and *n2* are number of records.

For more details, please refer to Chapter 6.

3.3.5. Proactively Protecting against Disk Failure

The proactive protecting against disk failure experiment uses the HDD SMART dataset. Before investigating the SMART data, I clean the dataset by removing those SMART attributes that have no variation during the monitoring period, and the attributes with high fluctuations which do not store the status of history points, because they do not contribute to understanding the dynamics of disk health. To avoid bias to any attribute, I use “min- max normalization” method to normalize the values of each attribute to the range of [-1, 1].

$$x_{norm} = 2 \times \frac{x - x_{min}}{x_{max} - x_{min}} - 1$$

After cleaning the constant attributes, I have 10 SMART attributes left (Table 3.2). Among them, CPSC, RUE, SER, HFW and HER show small variation, while RER, TC, SUT, POH and RSC display medium to large variation. To discover the type of disk failures and model the corresponding degradation process, it is important to find the critical SMART attributes that have strong correlation with the occurrences of disk failures. For each failed disk, I calculate the pairwise correlation between an attribute and the rest of attributes, and select the critical attributes which are highly correlated with other attributes (i.e., with a correlation higher than a threshold, such as 0.75). Besides, in order to understand the logical failure and predict future failure of the HDD, I also leverage the clustering machine learning algorithms, regression models, and some statistical methods to investigate the HDD SMART dataset. The Chapter 7 provides more details of it.

CHAPTER 4

RELIABILITY CHARACTERIZATION OF SOLID STATE DRIVES IN A SCALABLE PRODUCTION DATACENTER

In this chapter, I explore the SSD-specific SMART dataset to conduct an in-depth analysis of SSD reliability in a production environment. In this datatset, SSDs are used as a data buffer between the main memory and the storage subsystem.

4.1. SSD Caching and Flash Storage

Server-side flash or *SSD caching* refers to the deployment of SSDs as flash memory for caching and tiering data between the main memory and the storage system. As a cost-effective alternative to flash storage, it is often coupled with slower HDDs to improve the read and write throughput. When using SSDs as read cache, compute nodes retrieve data from permanent storage (HDDs) or via a storage area network (SAN), and store temporary copies of active data on NAND flash memory. Thus, data can be accessed quickly when needed. When used as write cache, SSDs buffer data until the slower and persistent storage has space and bandwidth to complete write operations. SSD caching is managed by system's storage controllers and is secondary to the main memory. Because the footprint of active data is relatively small, the capacity requirement of SSDs is lower than that of a full flash storage. *Flash storage* is getting popular in high performance storage appliances that use flash memory based technologies as the permanent storage media. It has higher capacity and reliability requirements, while the access frequency is usually less than that of SSD caching. In this research, I study the reliability of SSDs as caching/buffering devices.

4.2. Related Works

Many studies have investigated the bit error failure behavior of multi-level cells (MLC)[35, 33, 34] and single-level cells (SLC)[65]. They find that the bit error rate of the flash memory increases with an increased number of Program/Erase (P/E) cycles. These studies model the bit error rate as an exponential function of the number of P/E cycles that a cell has gone through. There are also a number of recent studies analyzing the statistical distributions of SSD failures in the field[63, 77]. They also find that although flash drives offer a lower field replacement rate than HDDs, they have a significantly higher rate of uncorrectable errors that can impact the stored data.

However, the existing studies of SSD reliability are either at the circuit level (i.e., MLC and SLC) or for the entire storage system level using field data. They do not explore the rich set of performance and reliability related attributes provided by the SSD SMART at the drive level. Compared with the SSD failure field data which do not provide insight into how SSD deteriorates and what factors dominate the process or workload and environment data which complicate SSDs' failure analysis, SSD-specific SMART data provide a direct and insightful way to characterize SSD failures with a generic method.

On the other hand, many existing works study the reliability of raw flash chips. Their evaluations are performed in controlled lab environments with only a limited number of models and devices. In general, they use synthetic benchmarks to stress individual flash components, and identify error symptoms and sources. For example, [31, 54, 53, 65, 35, 33] found that flash reliability is attributed to *read disturb error* and *program disturb error* which are caused by the tunneling effect where data in the untouched blocks are affected by read or program operations in the surrounding blocks. *Data retention error* is caused by detrap current that erratically changes the data at threshold voltage[94, 95, 29, 56]. The error prediction and recovery methods are discussed in [64]. The reliability of flash cells deteriorate over a number of P/E cycles [39, 80]. In [49], the cost, performance, capacity, and reliability trend of flash memory are studied. In the controlled environment, tests focusing on certain aspects of flash memory aim to eliminate unwanted effects. Results from these works provide a knowledge base on flash reliability, and are complementary to my work.

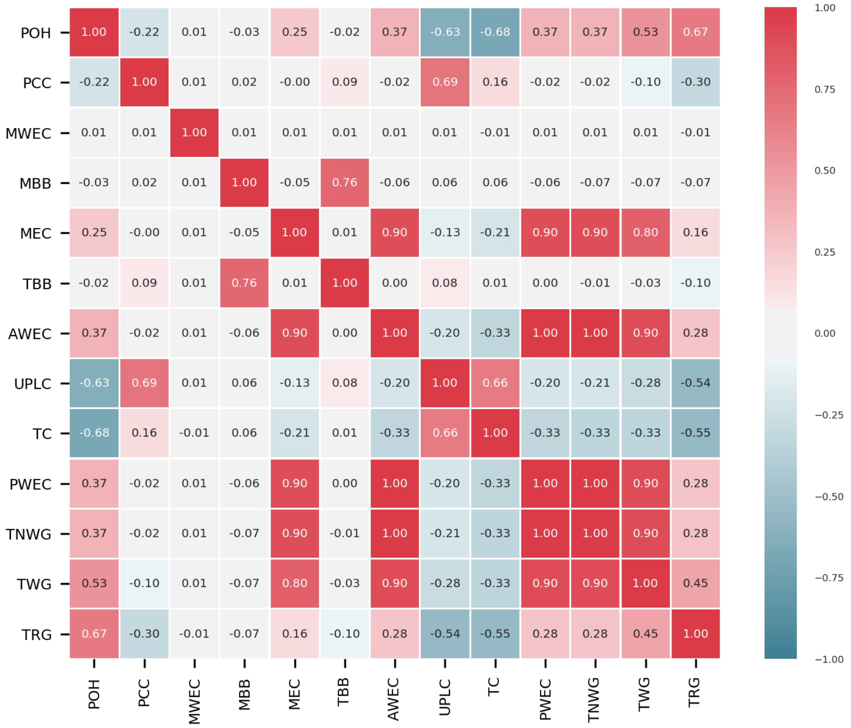
The aforementioned studies provide insights to chip-level flash reliability. It is also urgent to understand flash reliability in large-scale datacenters under real-world workloads. Recent works from Facebook[63], Google[77], and Microsoft[67] study field datasets. Their studies discover important differences in the field compared with those in the controlled environments. SSDs in their studies are used as permanent storage devices. In contrast, SSDs in the datacenter that I study are used as caching devices. In addition, my dataset includes six months of detailed SSD-specific SMART records, which are valuable for characterizing SSD reliability at the device level with rich semantic information.

4.3. Correlations Analysis of SSD SMART Data

Figure 4.1 and Figure 4.2 shows the pair-wise correlation calculated by using Spearman coefficients for the two SSD models in my dataset. Values of the Spearman corrleation coefficients range from -1 (i.e., strong negative monotone correlation) to $+1$ (i.e., strong positive monotone correlation), while 0 indicates no correlation. I compare the correlations of environmental attributes with workload-related attributes, and also with error-related attributes. I show the results with over 95% of confidence. In the correlation heat-map, redder colors indicate stronger correlations. Note that the solid red blocks along the diagonal show the correlation of an attributes with itself, which is not considered in my analysis. Other blocks with a correlation coefficient greater than a threshold, say 0.9, infer that the corresponding attribute pairs are significantly correlated. The major findings on the correlations of SSD SMART attributes are as follows.

4.3.1. Finding 1: Environmental Attributes Barely Affect SSD Reliability

After removing constant-valued attributes, I use 13 SMART attributes to calculate the correlation coefficients for Model-A SSDs, and 12 attributes for the Model B. Among these attributes, environmental attributes, i.e., Temperature in Celsius (TC), Power On Hours (POH), Power Cycle Count (PCC), and Unexpected Power Lost Count (UPLC), do not possess strong correlations with other attributes. If the threshold for the correlation coefficient is set to 0.6, the correlation between the following attributes needs analysis. Power On Hours (POH) and the Number of NAND Page Write by FTL (NONPWNTF) for Model-B SSDs have a correlation of 0.84, 0.67 between POH and Total Read in GB (TRG) for Model A, and 0.53 between POH and Total Write in GB (TWG). This is because older SSDs (i.e., higher POH values) are more likely to experience more read/write/erase operations. Thus, *environment-related SMART attributes do not significantly influence SSD reliability*, which confirms with prior studies [63, 77].



	POH	PCC	MWEC	MBB	MEC	TBB	AWEC	UPLC	TC	PWEC	TNWG	TWG	TRG
POH	1.00	-0.22	0.01	-0.03	0.25	-0.02	0.37	-0.63	-0.68	0.37	0.37	0.53	0.67
PCC	-0.22	1.00	0.01	0.02	-0.00	0.09	-0.02	0.69	0.16	-0.02	-0.02	-0.10	-0.30
MWEC	0.01	0.01	1.00	0.01	0.01	0.01	0.01	0.01	-0.01	0.01	0.01	0.01	-0.01
MBB	-0.03	0.02	0.01	1.00	-0.05	0.76	-0.06	0.06	0.06	-0.06	-0.07	-0.07	-0.07
MEC	0.25	-0.00	0.01	-0.05	1.00	0.01	0.90	-0.13	-0.21	0.90	0.90	0.80	0.16
TBB	-0.02	0.09	0.01	0.76	0.01	1.00	0.00	0.08	0.01	0.00	-0.01	-0.03	-0.10
AWEC	0.37	-0.02	0.01	-0.06	0.90	0.00	1.00	-0.20	-0.33	1.00	1.00	0.90	0.28
UPLC	-0.63	0.69	0.01	0.06	-0.13	0.08	-0.20	1.00	0.66	-0.20	-0.21	-0.28	-0.54
TC	-0.68	0.16	-0.01	0.06	-0.21	0.01	-0.33	0.66	1.00	-0.33	-0.33	-0.33	-0.55
PWEC	0.37	-0.02	0.01	-0.06	0.90	0.00	1.00	-0.20	-0.33	1.00	1.00	0.90	0.28
TNWG	0.37	-0.02	0.01	-0.07	0.90	-0.01	1.00	-0.21	-0.33	1.00	1.00	0.90	0.28
TWG	0.53	-0.10	0.01	-0.07	0.80	-0.03	0.90	-0.28	-0.33	0.90	0.90	1.00	0.45
TRG	0.67	-0.30	-0.01	-0.07	0.16	-0.10	0.28	-0.54	-0.55	0.28	0.28	0.45	1.00

FIGURE 4.1. Spearman Correlation among Attributes of Model A SSDs

4.3.2. Finding 2: Workload-related Attributes Do Not Directly Indicate Occurrences of SSD Failures

Flash cells can endure a limited number of program and erase (PE) cycles. I/O workloads could provide useful information about the wear level of flash cells. Early research reported an exponential growth of the Raw Bit Error Rate (RBER) with the increase of PE cycles [65, 37, 38]. However, recent field studies show the contradictory results, that is the increase of RBER is linear[77].

In Figure 4.1, I observe that those attributes that are related to write and erase operations of SSDs, such as Max Erase Count (MEC), Percentage of Write Erase Count (PWEC), Average Write Erase Count (AWEC), Total NAND Write in GB (TNWG), and Total Write in GB (TWG), have significant correlations which are higher than 0.9 between each other. However, the dataset does not show any correlation between these wear-related attributes with failure symptoms such as the Raw Bit Error Rate and Bad Blocks. Since the dataset contains six months of SSD SMART records, it is possible that flash chips have not experienced failures during that period of time.

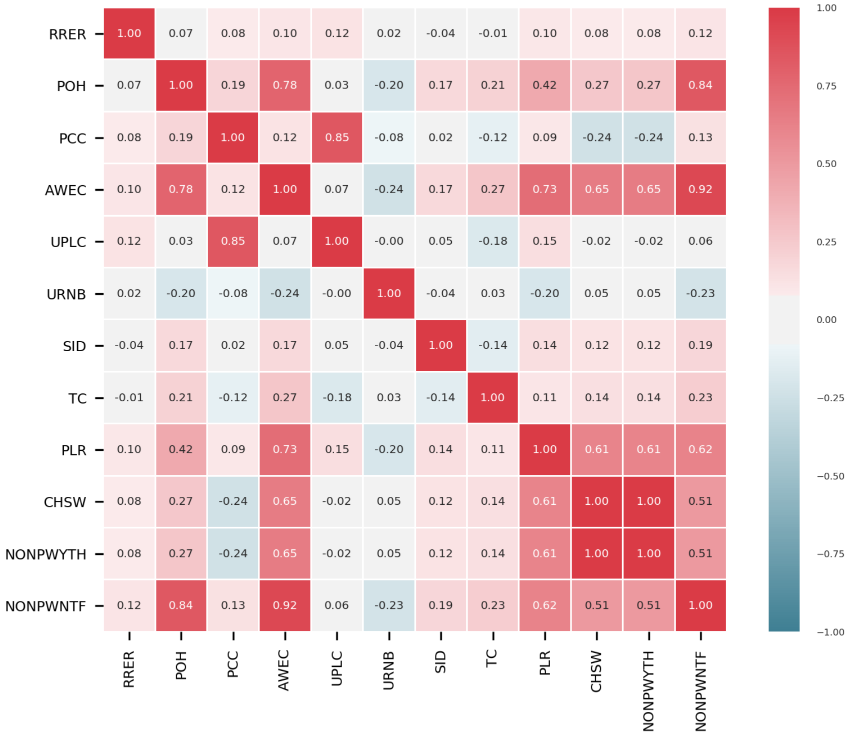
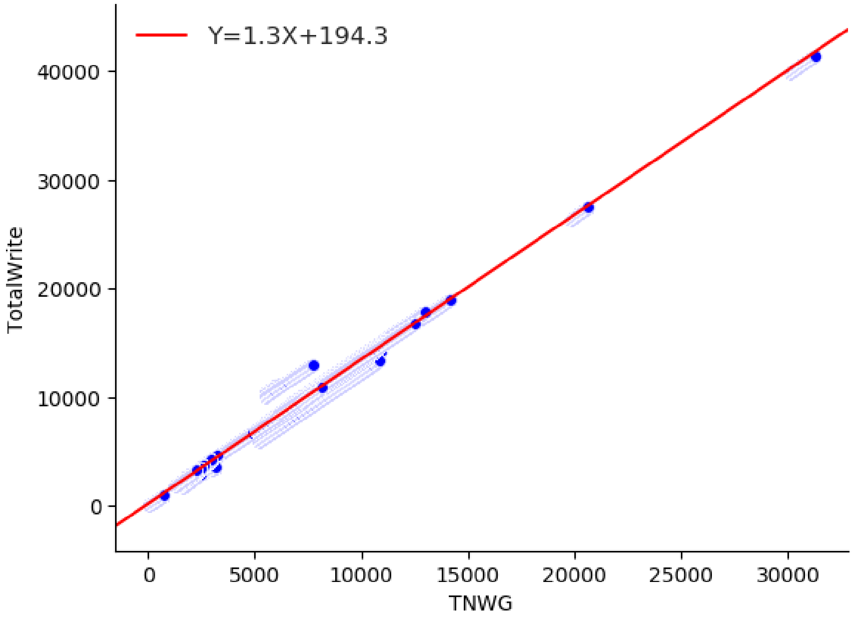


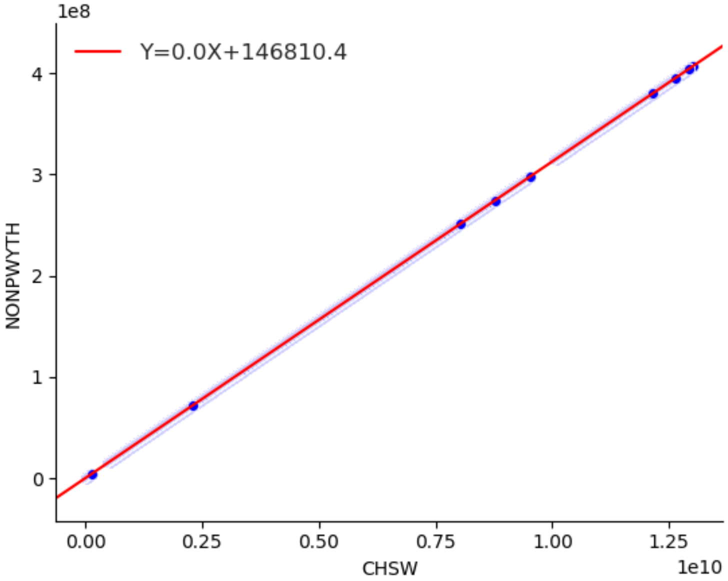
FIGURE 4.2. Spearman Correlation among Attributes of Model B SSDs

The raw values of TNWG and TWG of Model A, as well as the Number of NAND Page Written by Host (NONPWYTH) and Number of NAND Page Written by FTL (NONPWNTF) of Model B can be used to calculate flash memory's write amplification ratio. Figure 3(a) shows the linear regression that fits the correlation data between TNWG and TWG.

I observe a 1.3X write amplification from host initiated writes to the actual NAND page writes by FTL. Note the ideal write amplification is 1X. *The workload-related attributes do not directly indicate the occurrences of SSD failures.*



(a) Write Amplification



(b) Host writes vs. NAND Writes

FIGURE 4.3. Relationship of Write Operations.

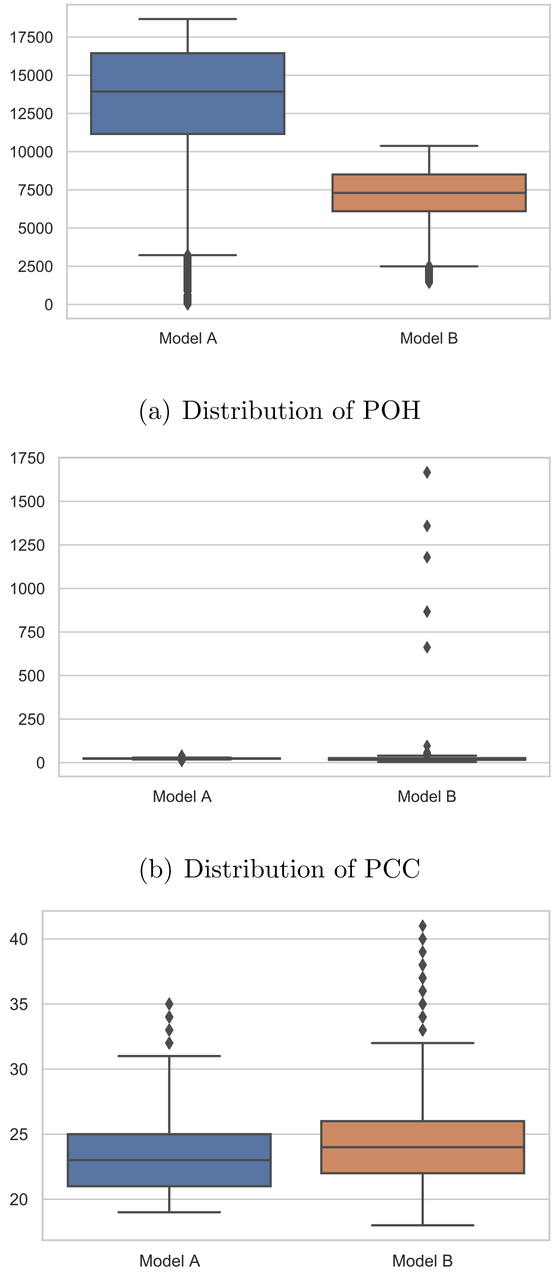
A similar pattern is observed for Model-B SSDs. As illustrated in Figure 4.2, Average Write Erase Count (AWEC) has a strong correlation with Percent Lifetime Remaining

(PLR), Cumulative Host Sectors Written (CHSW), Number of NAND Page Written by Host (NONPWYTH), and Number of NAND Page Written by FTL (NONPWNTF). Among these attributes, PLR indicates the estimated percentage of lifetime remaining based upon the average number of block erase operations and the number of rated block erase operations. The average number of block erase operations has a positive correlation with AWEC. Henceforth, PLR becomes positively correlated with AWEC. I also find that 90.8% of SSDs in the datacenter have PLR equal to zero, which means those SSDs reach the end of their lifetime according to the specification of PLR. However, the error-related SMART attributes for those SSDs show no significant difference from other SSDs whose PLRs are greater than zero. In addition, those SSDs run smoothly for a long period of time with PLR remaining as 0%. This finding also confirms that manufacturers' rated block erase count is conservative, and SSDs' actual lifetime in field is longer than that provided by the manufacturers.

The results also show that Cumulative Host Sectors Written (CHSW) and NAND Page Written by Host (NONPWYTH) have a correlation coefficient of 1.0. CHSW indicates the amount of data that the host writes to the LBA device. FTL then translates and maps the LBA sector requests to physical pages on an SSD. The number of pages written is recorded by NONPWYTH. The number of bytes written to the SSD recorded by the two attributes should be the same. A typical sector size is 512 bytes, and the page size of Model-B flash memory is 16 KB. This corresponds to my observation that the mapping from sectors to pages has a ratio around 30:1 as shown in Figure 3(b). The correlation results between PLR and error-related attributes from Model-B SSDs also confirm that *not every workload-related SMART attribute, such as PLR, directly indicates that SSDs fail.*

4.3.3. Finding 3: I/O Workloads Are Not Evenly Distributed in The Datacenter

I also investigate the distributions of environmental attributes and their strongly correlated attributes. The cross-comparison identifies two models for workload, wear level, and cumulative failure symptoms. In this study, I use boxplots to illustrate the attributes' distributions for ease of visualization and analysis. Figure 4.4 shows the environmental attributes from both SSD models with outliers, while Figure 4.5 shows the environmental attributes from both SSD models without outliers. The boxplot with outliers show a few significant large PCC values from Model B. Thus, to better understanding distributions, Figure 4.5 eliminates the outliers.

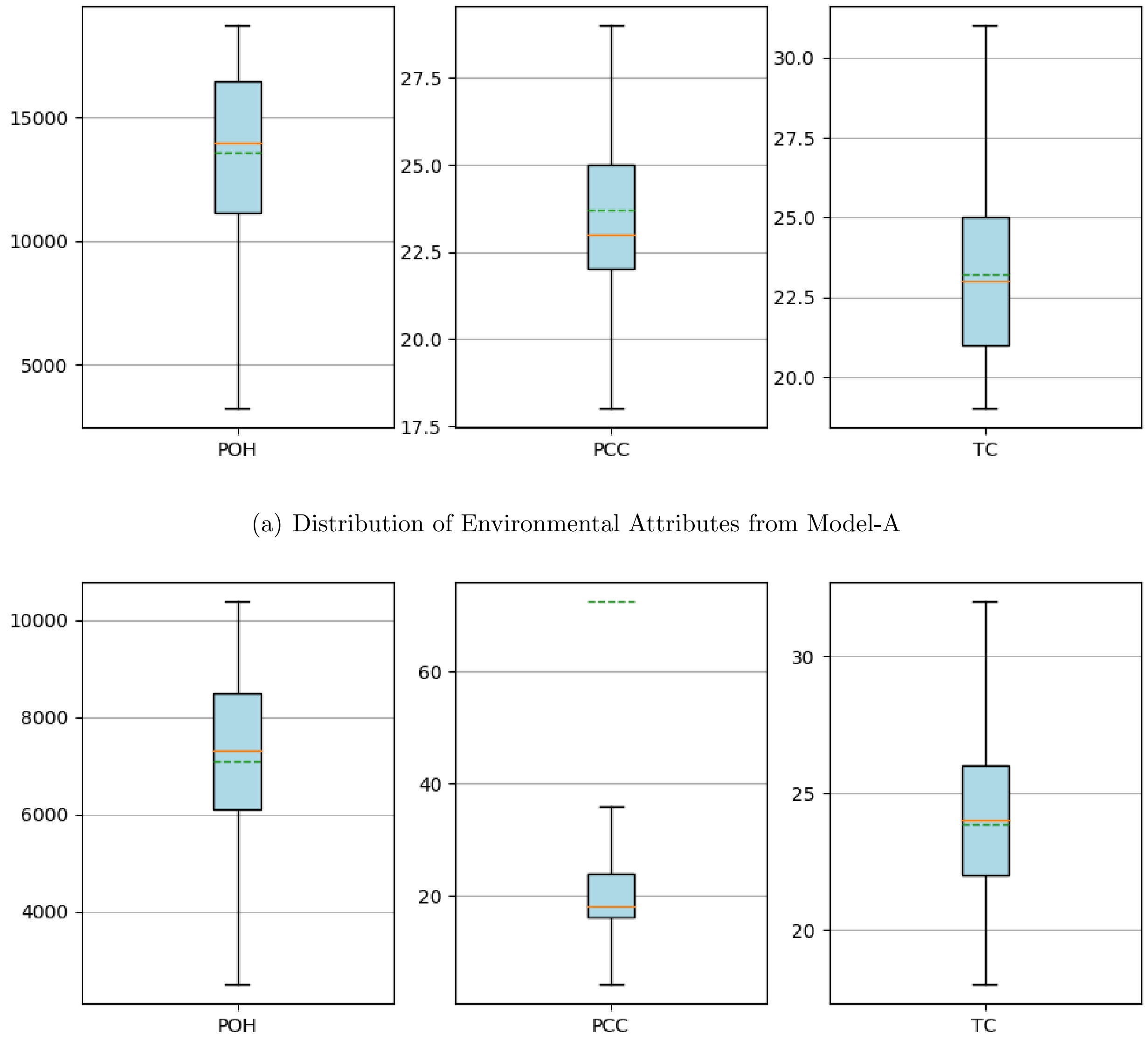


(a) Distribution of POH

(b) Distribution of PCC

(c) Distribution of TC

FIGURE 4.4. Distributions of Environmental Attributes from Model A and Model B – With Outliers.



(a) Distribution of Environmental Attributes from Model-A

(b) Distribution of Environmental Attributes from Model-B

FIGURE 4.5. Distributions of Environmental Attributes from Model A (Top) and Model B (Bottom) – Without Outliers.

From Figure 4.5, the box region shows the first quartile (Q1) to the third quaritile (Q3) of raw values for environmental attributes, that is Power On Hours (POH), Power Cycle Count (PCC), and Temperature Celsius (TC). In these figures, where IQR is the interquartile range (Q3-Q1), I set the whiskers of boxplots using the default value 1.5. Thus, the upper whisker shows the maximum value at (Q3+1.5*IQR), where the lower whisker shows the minimum value at (Q1-1.5*IQR). Beyond the whiskers, data are considered outliers and do not show in figures. Solid and dash lines represent the median value and arithmetic mean of each attribute respectively. I analyze the median values in the following discussion as medians are not skewed so much by extreme values, for example, PCC value in model B. And so medians give a better idea of typical values to compare between SSD models.

The median TC value for both SSD models is relatively close. This is because 1) the cooling system in the datacenter functions well, and 2) the internal thermal management of SSDs keeps drives under a stable temperature.

At the same time, I observe that Model-A SSDs have about two times longer operation time than Model-B drives based on the POH. The variance of POH is high, ranging from 2,000 to 19,000 for Model A, and from 2,000 to 11,000 for Model B. Considering the median PCC for both models is close (i.e., around 20-22), I consider that the value of POH is not solely determined by the operation time. After checking manufacturers' documents, I find that the raw value of POH reflects a device's online hour (i.e., under power), and excludes the increment in offline states such as *SATA Partial*, *SATA Slumber*, and *SATA Device Sleep*. I also consulted with system administrators working at the datacenter who confirmed that both SSD models in the system were deployed in the same time frame. Despite less than 5% of the SSDs experienced infant mortality and were replaced, the majority of SSDs were operated in an active system since their deployment. Since enterprise datacenters employ more aggressive power saving policies, I believe that Model-B SSDs have experienced less workload as the time spent in offline states is about two times more than Model-A SSDs. I also notice that this workload imbalance not only happens between different models of SSDs, but also among drives of the same model. The difference between PCC median and PCC mean in model B is caused by a few drives (¡10% of SSD population) that have very high PCC values. All these observations show that the *I/O workloads are not uniformly distributed among SSDs.*

In summary, I conclude the following 4 findings: 1) write and erase operations have a strong correlation between each other; 2) environmental attributes do not directly affect

SSDs' health; 3) workload-related attributes do not directly indicate the occurrences of SSD failures; 4) I/O operations are not evenly distributed in the system.

4.4. Characteristics of SSD Reliability

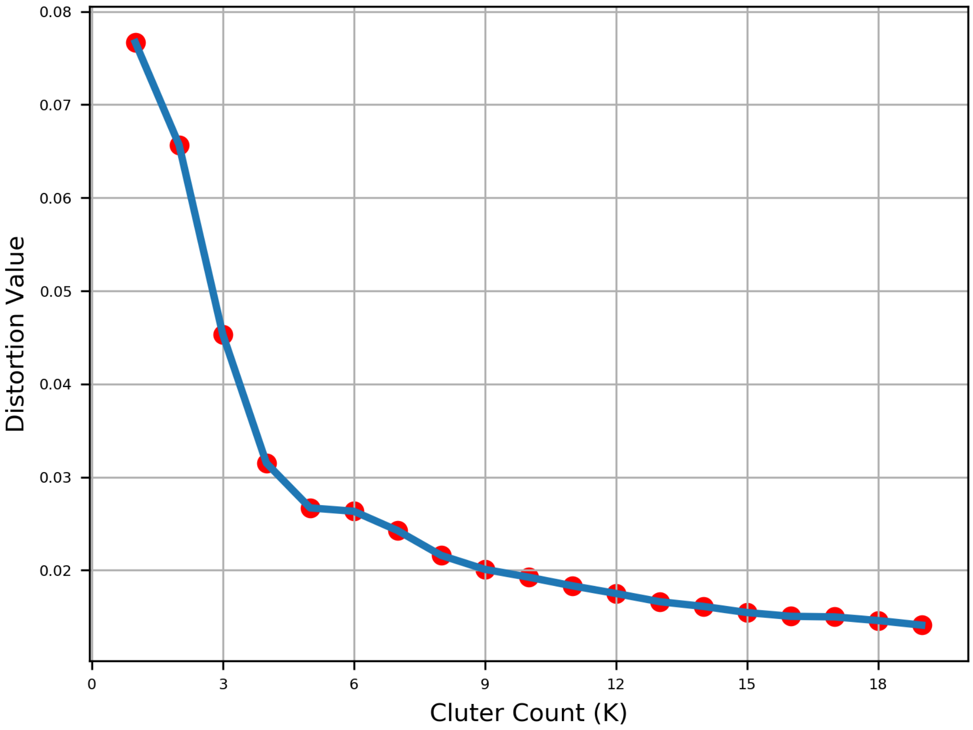


Figure 4.6. Model A SSD Elbow

I use "elbow method" to choose 5 as optimal number of clusters for k-means clustering. SMART attributes MEC, WEC, PWEC, TNWG and TWG are used in clustering.

Based on the material composition and architecture of SSDs, I/O operations, including read, write and erase, can influence the health status of SSDs. We analyze the categories of SSD health states and their possible transitions over time. Experimental results show that the SSD SMART records can be grouped into five clusters for both Model-A and Model-B SSDs, which is shown in Figure 4.6 and Figure 4.7. For Model-A SSDs, SMART attributes MEC, WEC, PWEC, TNWG and TWG etc. are selected for clustering, while AWEC and NONPWNTF, CHSW and NONPWYTH etc. are used to cluster SMART records for Model

B. Due to the high dimensionality, we do not plot the five clusters. With the euclidean distance cost function, Model-A SSDs have a distortion value lower than 0.03, while Model-B drives have a lower than 0.01 distortion value.

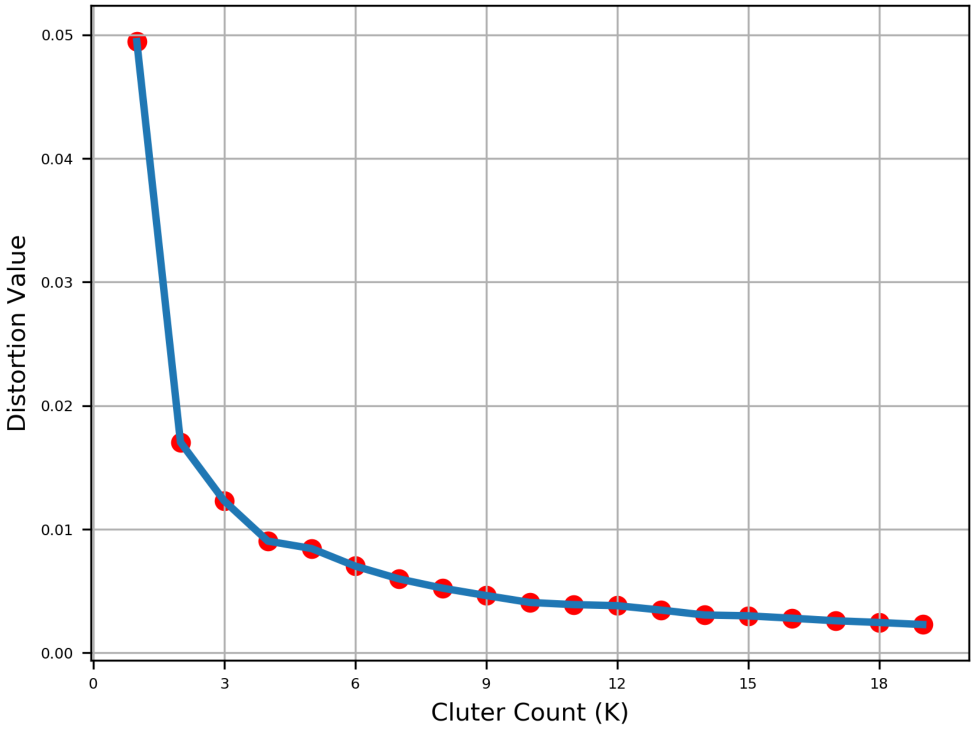


Figure 4.7. Model B SSD Elbow

I use "elbow method" to choose 5 as optimal number of clusters for k-means clustering. SMART attributes AWEC and NONPWNTF, CHSW and NONPWYTH are used in clustering.

An important finding from my experimental results is that, for both models of SSDs, the health states of SSDs may change from one group to another as the wear level changes. In several cases, such transitions happen more than once. In my study, a maximum of three transitions is observed for Model-A SSDs and a maximum of nine transitions is observed for Model-B SSDs. For Model A, over 84% of SSDs experience health state transitions, which called *reliability degradation*. Model B SSDs also have over 36% drives experience health state transitions. Table 4.2 and Table 4.1 present the relative size of each SMART record cluster and the frequency of reliability degradation. From the tables I can see the patterns of reliability degradation between the two models of SSDs are different. Specifically, the majority of Model-A SSDs have reliability degradation, while Model-B SSDs have more stable health states. As I discuss before that I/O workload is unbalanced between Model-A and Model-B SSDs, those drives of Model B with less workload are more likely to stay in one health state.

Table 4.1. Groups of SSD SMART Records and Transitions of SSD Health States (Model A).

CATEGORIES	CLUSTERS	PERCENTAGE of SSDs(Model A) in CATEGORY(%)
Cluster	Cluster 0	10.7%
	Cluster 1	0.0%
	Cluster 2	1.3%
	Cluster 3	2.0%
	Cluster 4	2.0%
Cluster Transition	Cluster 1→4	27.3%
	Cluster 3→1	18.0%
	Cluster 3→1→4	38.0%
	Cluster 3→1→4→0	0.7%

To analyze the wear levels and relationship between SMART records and SSD health states, I investigate each cluster produced by K-Means. I find each SSD model has its own reliability characteristics and has some properties in common. The distributions of the selected attributes among different SSD groups are shown in Figure 4.8. For Model-A SSDs, I find that 1) drives in Cluster 2 experience I/O intensive operations. The number of read operations is the highest, and the number of write and erase operations (as shown by 5Combine in Figure 4.8) is lower than those in other clusters. 2) Drives in Cluster 0 experience the highest number of write and erase operations, while the number of read operations is the average. (3) Clusters 1, 3 and 4 include the majority of drives which experience the average number of I/O operations. However, the reliability degradation of SSDs in the three clusters follows similar transition patterns, i.e., Cluster 3 → Cluster 1 →

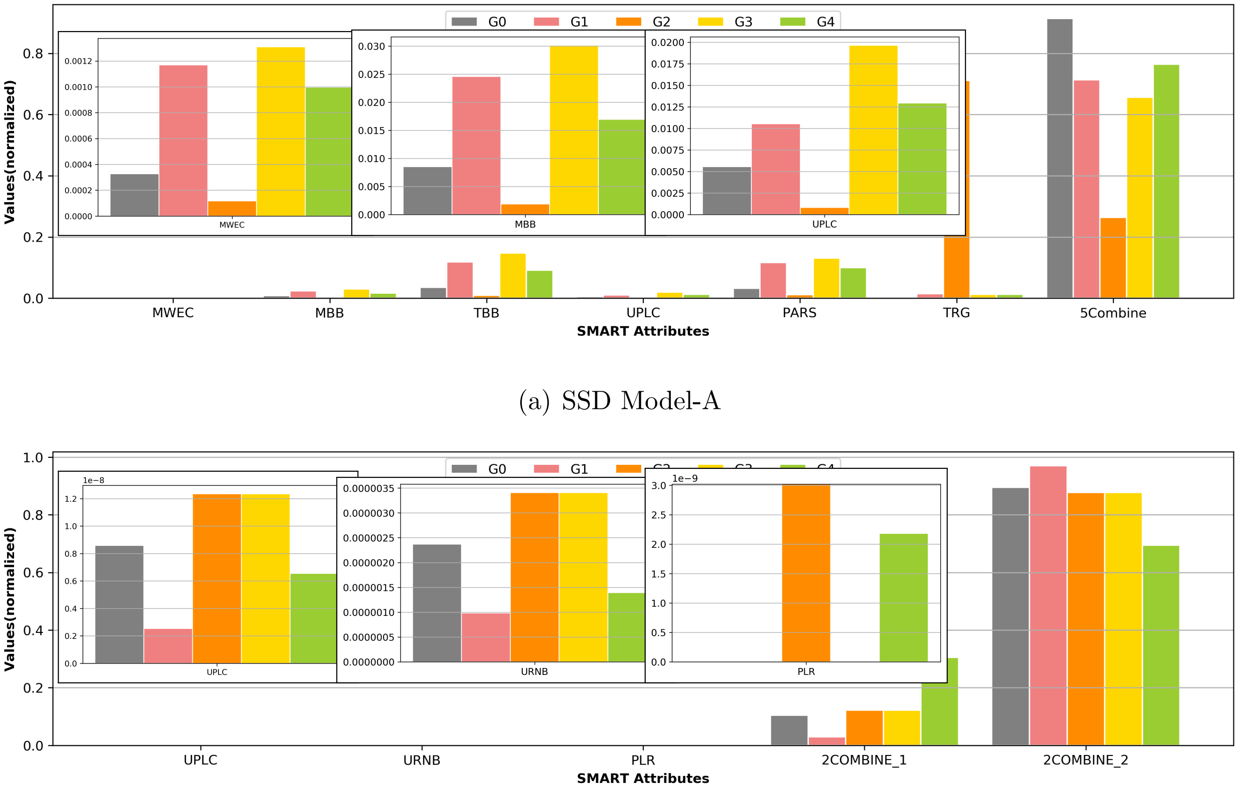
Cluster 4. Along this transition, the value of Total Bad Block (TBB) decreases while the number of write and erase operations increases. Based on the preceding findings, I infer that the health states of SSDs in Clusters 2 and 0 is worse than other drives which are still in good shape. Those good SSDs will experience reliability degradation, i.e., transition to Clusters 1 and 4, as more I/O operations and P/E cycles cause wear and errors.

TABLE 4.2. Groups of SSD SMART Records and Transitions of SSD Health States (Model B).

CATEGORIES	CLUSTERS	PERCENTAGE of SSDs(Model B) in CATEGORY(%)
Cluster	Cluster 0	19.8%
	Cluster 1	27.0%
	Cluster 2	0.7%
	Cluster 3	15.6%
	Cluster 4	0.7%
Cluster Transition	Cluster 3→0	29.8%
	Cluster 0→3	0.7%
	Cluster 3⟷0	4.3%
	Cluster 3→0→1	0.7%
	Cluster 1→0→3	0.7%

For Model-B SSDs, I find that 1) drives in Cluster 0, 2 and 3 experience similar write and erase operations for both FTL and host; 2) drives in Cluster 4 experience the highest number of FTL write operations (as shown by 2COMBINE_1 in Figure 4.8), while drives in Cluster 1 experience the lowest number of FTL writes; 3) for Host write operations (i.e., 2COMBINE_2 in Figure 4.8), drives in Cluster 1 experience the highest number while those in Cluster 4 have the lowest Host writes; 4) counter-intuitively, PLR cannot indicate the remaining lifetime of an SSD. Only SSDs in Clusters 2 and 4 have PLR > 0, while PLR of SSDs in other clusters remains 0; 5) Clusters 3 and 0 have the majority of drives (that is 70.2% as see in Table III). A half of the drives experiences reliability degradation. Among them, 85.8% of SSDs follow a similar degradation pattern, that is Cluster 3 → Cluster 0.

Drives in Cluster 3 have more unused NAND blocks than those in Cluster 0. Based on the preceding findings, I believe that SSDs in Clusters 0, 2 and 3 experience the similar write workloads. SSDs in Cluster 3 are in a better health state than those in Cluster 0.



(a) SSD Model-A

(b) SSD Model-B

Figure 4.8. Attributes' Values of Cluster Centroids (Top–Model A and Bottom–Model B).

For both models, I/O operations affect SSDs' health status, and even cause reliability degradation of the drives. Workload-related attributes play an important role for SSD reliability analysis. As a characteristic of SSD reliability, I discover that the reliability degradation of SSDs follows certain patterns which depend on the model of a drive and I/O workload that the drive performs.

CHAPTER 5

AN EMPIRICAL STUDY OF QUAD-LEVEL CELL (QLC) NAND FLASH SSDS FOR BIG DATA APPLICATIONS

In this chapter, I would like explore and evaluate the performance, as well as reliability of QLC SSD in modern data centers. Cause QLC is a such new technology that no data center apply it in large scale, I set up the experiment in lab servers.

5.1. Related Works

QLC SSD is a new product targeted at read-intensive workloads for use in datacenters. As far as I know, there are only two related research papers focus on this aspect. The research conducted by Yoshiki et. al [84] focuses on QLC NAND flash memory power consumption and performance analysis on different heterogeneous SSD configurations. Their research points out that SCM(Storage Class Memory)/TLC configuration is optimal for cold workloads; while SLC/QLC configuration is recommended for hot workloads. Another research is purposed by Liu et. [58]. This chapter studies efficient coding methods for QLC NAND flash. Their paper presents four enhanced Gray codings to QLC NAND to improve efficiency for read operations and data error correction. To distinct my work from the previous researches, I emphasize the performance evaluation of QLC SSD as a contender for HDD and other types of SSDs in datacenter storage systems. My evaluation also compares QLC SSD against MLC or TLC SSD in terms of the economic aspects and analyzes how QLC SSD will change the landscape of modern datacenters.

The 3D NAND QLC is by far the most promising solution to achieve the “high capacity, high reliability , low cost” goal in SSD storage. But it is not the only solution. Another famous storage technology is called *3D XPoint* [18] by Intel. Intel integrated this technology in its *Optane memory*, as well as applying this technology to its SSDs, namely *Optane SSD*. Micron also has its own 3D XPoint brand, named *QuantX*. But Micron does not has any SSDs available that come embedded with this technology. The performance evaluation shows that *3D XPoint* SSD achieve better write latency and I/O speed than most 3D NAND SSDs in the market, according to the research [12]. However, the price of *Optane* SSD is still 4-5X greater.

5.2. Evaluating Performance and Value of QLC SSD for Modern Data Centers

TABLE 5.1. Types of SSD in Evaluations

Brand Name[*]	**Cell Type**[+]	**Architecture**	**Capacity (GB)**	**Cost per bit**
Brand A	eMLC	2D	240	$$$
Brand B	TLC	3D	480	$$
Brand C	QLC	3D	480	$
Brand D	eQLC	3D	1920	$

[*] For each brand, the logical sector size are 512 byte; physical sector size may vary.

[+] The "e" in front of a cell type denotes the enterprise grade drive.

5.2.1. Experiment Setup

Table 5.1 highlights the main features of each SSD used in my experiment. My evaluation comprises of several factors that might impact SSD performance. All of my experiments are performed on two HP Proliant ML110 G6 Storage servers with identical configuration. Each server is equipped with an eight core Intel Xeon (3 GHz), 8 GB DRAM, and Ubuntu 18.04 LTS. All HDDs and SSDs are physically attached to the sever machine via SATA 3.0 connectors. I use the `fio` (aka., Flexible I/O) synthetic trace to simulate various types of workloads. During the experiment, `fio` was set to use asynchronous engine for non-buffered I/O, and the I/O depth were set to 64 to saturate the bandwidth. The broad range of factors that might affect the performance of SSD in production environment includes read-write ratio, data access patterns, block size, garbage collecting operations, bad block managements and reserved block replacement policy, etc. The total workload size exceeds available memory to ensure a storage-centric workload. I repeat each test five times then report the average.

Note that new SSD needs to break-in before the experiments. Since brand new SSDs shipped with empty flash blocks, I/O latency measured at empty blocks will differ from non-empty blocks. The break-in process fills the new drive with nonzero data. I/O performance measured from non-empty block represents real-world results from production environment.

5.2.2. Performance Evaluation

In production storage systems, different applications exhibit distinct I/O patterns and characteristics. We can categorized them into two types: small reads/writes and large reads/writes. The former is typically measured by IOPS, while the latter is evaluated by throughput. In my preliminary testing, I adopted the widely used benchmark configuration and procedures to evaluate the performance of an SSD. I selected the following three metrics to quantitatively measure the performance. The objective of each metric is highlighted as follows.

(1) **Sequential Write/Read with 1MB block size.** This test measures I/O bandwidth for large I/O requests. In this test, sequential write/read are performed in multiple parallel streams, using 1MB I/O size to simulate large data writes/reads.

(2) **Write/Read IOPS with 4KB block size.** This test measures the ability of a block device to handle small I/O requests. Following the industrial best-practice, I set I/O size to 4KB. Write/read are only performed in single stream, so the number of concurrent request is adjusted to a larger number to generate sufficient requests before they saturate the I/O bandwidth.

(3) **Write/Read Latency with 4KB block size.** This test evaluates the latency of a block device completing a I/O request. The write/read are performed in single stream and I/O array size is set to a small number, so the number of concurrent request is adjusted down to prevent reaching the maximum bandwidth or maximum IOPS.

Table 5.2. Benchmark Results

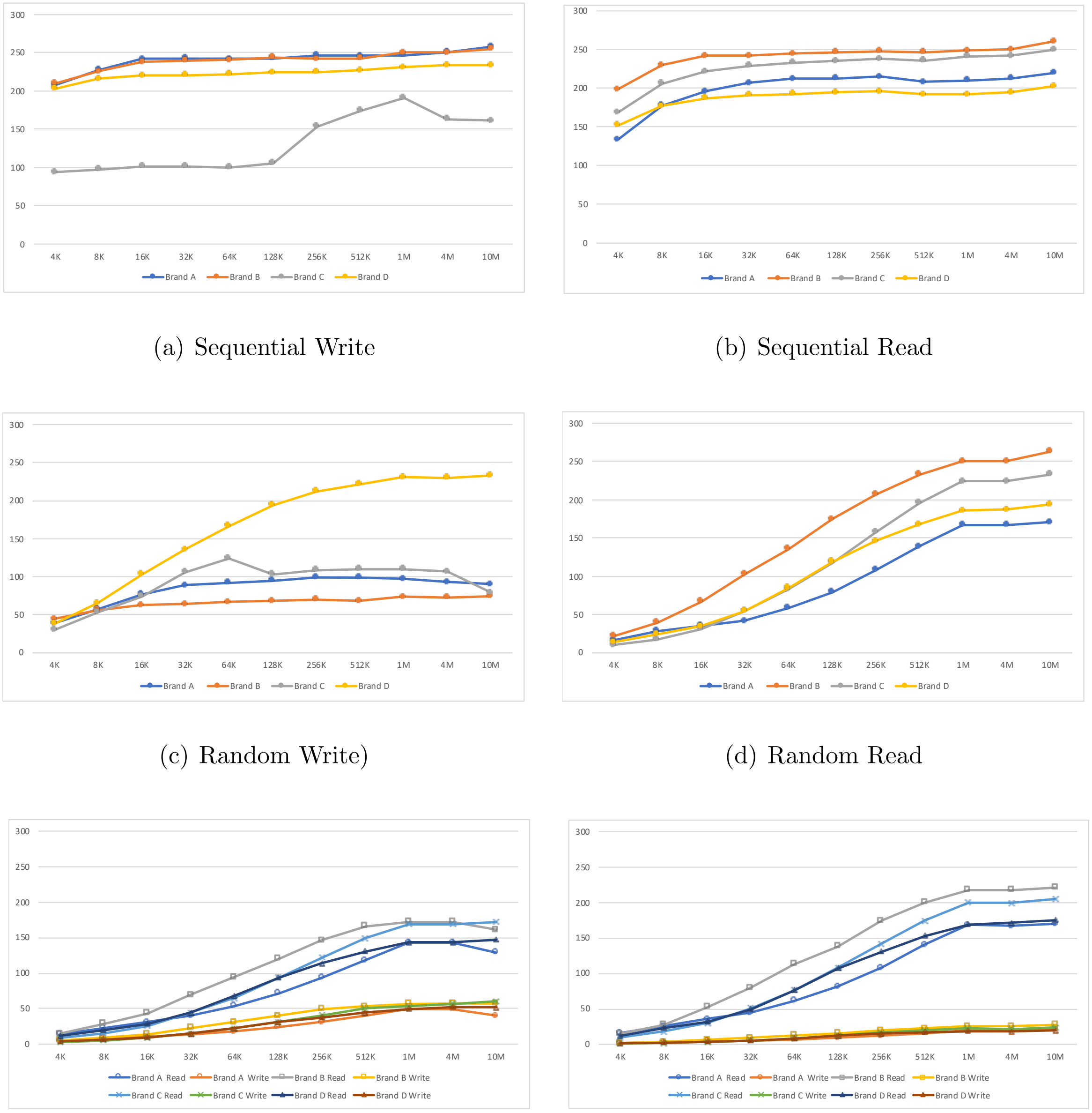
Metrics	**Brand A**	**Brand B**	**Brand C**	**Brand D**
Write				
Throughput (MB/s)	247	250	191	231
IOPS	9663	10400	6853	8852
Latency (μs)	406.1	378.9	574.7	446.3
Read				
Throughput (MB/s)	210	249	241	192
IOPS	4468	5337	2441	3338
Latency (μs)	896.9	763.4	1631.9	1195.1

Table 5.2 shows the preliminary results in these experiments. Overall, the write/read of both QLC drives performs worse than that of other drives. The brand C QLC drive has the lowest writing speed at 191 MB/s. Brand D QLC drive performs better at 231 MB/s. However, the write IOPS of both QLC drives can only achieve 66% - 90% of its MLC and TLC competitor, while the write latency are also 40μs-200μs higher than Brand A and Brand B drives. Similarly, the sequential read IOPS of QLC drives are 25% - 55% lower than its competitors and the read latency almost doubles the MLC and TLC drives. I cannot simply conclude that all QLC drives performs worse than TLC or MLC drives, but from the result I can extrapolate that SSD performance will degrade when the data bits per cell are increased. This result is intuitive as the increase of bit per cell require advanced architecture design and complicated electron level controls. In addition, QLC only have around 500 to 1000 P/E cycles. To prolong its lifespan, some QLC SSDs throttle the write performance by design.

On the other hand, QLC SSD packs 33% more data per cell (4 bits rather than 3) and adopts more sophisticated algorithms to encode data. Hence, they might exhibit different

I/O characteristic than industry best-practice for MLC and TLC drives. In the following sections, we explore a range of factors to optimize the storage configurations that yield better read performance for QLC SSDs.

5.2.2.1. Block Size Matters



(a) Sequential Write

(b) Sequential Read

(c) Random Write)

(d) Random Read

(e) Read and Write in 75/25 Ratio

(f) Read and Write in 90/10 Ratio

Figure 5.1. Block Sizes Effects on Read and Write

My preliminary test uses 4KB data block size. However, this block size may artificially inflate the total I/O number that the drives are capable of handling [19], and the I/O patterns in real-world scenarios are also more complicated. We may encounter different data block sizes with a mix of reads and writes requests. To better understand the QLC I/O characteristics for read-centric workloads, I test various block sizes with the read/write ratio at 75/25. The read and the write operations are also randomly mixed to simulate real-world scenarios. Figure 5.1 shows the experiment results. I increase the block sizes from 4KB to 10MB, and tested both sequential and random I/O operations. From the results, I have the following observations.

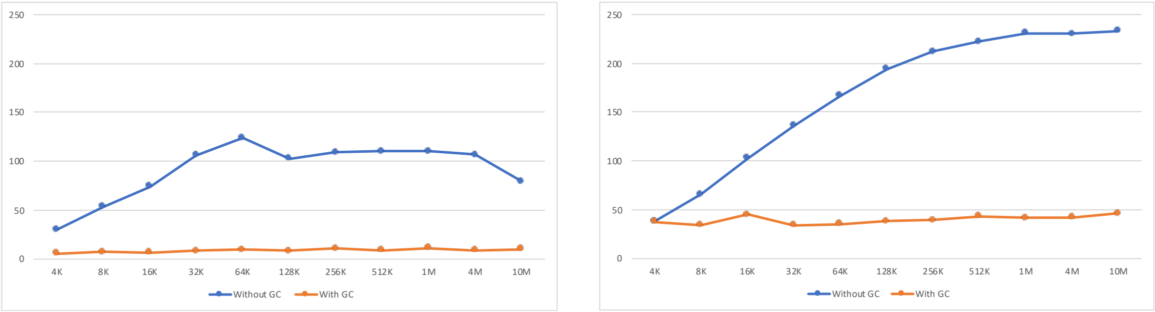
- **I/O speed increases with the block size.** Overall, the performance of sequential and random write/read operations of QLC SSD increases with the block size. I observed a similar trend in SLC, MLC and TLC SSDs. Since the logical block size of each drive is 512KB, write requests can only be handled per block unit, I believe the writing performance degradation, when block size exceed 1 MB, is due to data buffering or write aggregation. *I conclude that for the write process, SSD logical block size positively impacts the I/O performance.*
- **I/O speed increases faster when block sizes are smaller.** Its obvious that sequential I/O speed increases faster when the block size is less than 16KB. The random read performance increases rapidly until the block size reaches 1MB. However, for random write request, each drive have a different optimal block size. The experiment results also indicate that too many tiny files or massive files may hurt SSD performance. *The optimal block size should be in the range of 16KB to 1MB.*
- **Performance of enterprise-grade QLC SSD is more stable and predictable than consumer-grade QLC SSD.** In this experiment, I evaluated a consumer-grade QLC SSD (Brand C) and an enterprise-grade QLC SSD (Brand D). In both write and read tests, Brand D QLC SSD strictly follows the increasing trend as other MLC and TLC drives. But the I/O performances of Brand C SSD has more fluctuations, especially during the write tests. For the sequential write test, the

throughput of Brand C SSD peaks at 1MB block size before it starts to degrade. For a random write test, the throughput of Brand C SSD rapidly grows at first but then degrades after the block size exceeds 64KB. I still need further study to fully explain the fluctuating performance of Brand C SSD, but *I believe the SSD controller design has a major impact on I/O performance.*

5.2.2.2. Garbage Collection Matters

In SSDs, the garbage collection (GC) process releases the blocks that were occupied by invalid data. Recall that GC is usually performed at background when the drive is idle, so it minimize the performance impact while ensures the available drive capacity. Such a strategy is typically useful for consumer environment as they tend to have more idle time. However, enterprise environment have a much more intensive storage usage, causing the GC procedure to lack having sufficient time to perform its task in the background. When GC is eventually forced to run in the foreground alone with the application I/O payload, it imposes a significant performance and endurance impact to the system, especially for the write performance.

GC activities may have distinct performance impact for different SSDs, as it is effected by the embedded GC algorithm, the wear-leveling algorithm, the SSD controller policy, the amount of SSD empty blocks, capacity, block size, and other factors. Theoretically, QLC SSD needs to spend more efforts on GC than other types of SSD due to the complexity of NAND cells design. To measure the performance impact of GC for my QLC SSDs, I first fill up the SSD with random data then immediately issue burst I/O workloads. This will invoke the GC to release the invalid data blocks before handling new write request. Fig 5.2 shows the I/O performance difference between QLC SSD with GC and without GC in different I/O size. On average, Brand C SSD random write performance drops 90% when garbage collection onset, while Brand D SSD drops about 76%. Garbage collection activity not only impact write performance, but also affects read operations. Recent studies [83] found that read performance also degrades significantly when garbage collection is engaged; the read request will also be blocked until garbage collection process finish.



(a) Brand C QLC SSD

(b) Brand D QLC SSD

Figure 5.2. Garbage Collection Effects on Random Write (Throughput(MB) vs. Block Size)

5.2.2.3. Bad Block and Reserved Block Matters

SSD is a masterpiece of complex industrial products that comprises of thousands of sub-components. So, besides the block sizes and garbage collection, SSD performance might also affected by the number of bad blocks and reserved blocks. A block is marked as a bad block when its P/E cycle reaches a preset threshold or it becomes inaccessible. When a block is marked as a bad block, the SSD controller will map its logical address to a spare block from the reserved block area. Data in the inaccessible bad block is considered as lost since read and write requests cannot be completed. However, if the SSD supports block level redundancy such as Erasure coding or internal RAID, then the SSD controller can initiate the recovery procedure that reconstruct the lost data to the spare blocks. The recovery procedure will impact I/O performance as it requires additional I/O resources. Moreover, as the number of bad blocks accumulates, the number of reserved blocks will be exhausted. As a result, the available capacity of the SSD eventually shrinks. QLC SSD is more likely to encounter this problem as it has much lower P/E cycles.

5.2.2.4. Environment Matters

Like other types of SSD, environment factors such as power surge, radiation and operating temperature also impact the QLC SSD performance.

- Power: QLC SSD is a NAND flash based drive that are non-volatile. It can retain data even without power supply. However, study [97][86] shows that sudden power outage (and the associated power surge) will flip the bits and cause data corruption. Even in datacenter environment where power supply is generally stable, long power-on hours might lead to electron discharge and cause NAND flash to lose its data retention ability. When any bit errors are detected, the SSD controller will engage built-in error correction code (ECC) mechanisms to resolve the silently corrupted data. As a result, I/O performance will also be significantly affected by the ECC activity.

- Radiations: Cosmos radiation can disrupt the NAND flash cell energy level and causing soft errors. It can also permanently damage the semiconductor, leading to malfunction. Radiation problems not only occurs to SSD drives used in space flights but also impact datacenters at higher altitude.
- Temperature: Temperature has significant impact on the physical characteristics of NAND flash cell, hence indirectly affects the SSD's I/O performance. As the working temperature increasing, the oxide tunnel in NAND gates loses the ability to retain its charge level. Electrons will be able to escape from the tunnels much easier, which leads to bit flipping and soft errors. To tackle this problem, most SSD controllers implement thermal throttling mechanisms that artificially decrease the I/O resource quota when it detects temperature increase nearing the set threshold. This gives ECC mechanisms more time to correct the increasing number of corrupted bits. However, thermal throttling significantly degrades the SSD performance. When storage systems on heavy workloads or storage rack are placed at areas that has bad air circulation or ventilation, thermal throttling might be triggered much more frequently and will greatly impact the overall performance of the storage system.

5.2.3. Economic Analysis

With the ever-increasing data load and read-centric requests, datacenters are pressed to meet the increasing storage and service demands. When upgrading the storage infrastructure from the existing one or building a new one, IT professionals are constrained to the binary options of: performance-oriented 2.5-inch 10K RMP HDDs that offers higher performance but smaller capacity (e.g., 2.4TB), or 3.5-inch 7200 RPM HDDs that have higher capacity (e.g, 14TB) but lower read IOPS. Therefore, we can transitioning the standard twelve-bay 2U storage server into high-performance node that have 29TB raw capacity, or high-capacity node that have 168TB of raw capacity. QLC SSDs from major manufactures offers up to 7.6TB capacity (available in Brand D) per 2.5-inch drive that transitioning the same 2U server into 184TB raw capacity (i.e., 10% more than capacity tier HDDs). New QLC SSD offers a higher-density and higher-performance storage for the same datacenter footprints. In addition, QLC SSD has the following advantages that makes it beneficial to replace HDDs for read-intensive workloads (data listed in Table 5.3).

TABLE 5.3. Per Drive Cost vs Performance

Characteristics	**HDD**	**eMLC**	**TLC**	**eQLC**
Random I/O MB/s	50	167	250	186
Read IOPS	189	4468	5337	2441
$ / GB	0.02	0.67	0.2	0.12

- **Power efficient**: the HDD power consumption is around 8-11 watts while QLC SSD is around 3 watts (3X less). However, the read IOPS per watts shows QLC SSD has 38X higher power efficiency for the real-world power consumption.
- **Reliable**: the maximum data bytes that passed through the HDD drive interface (both read and write) is typically 2.5 - 3.5 petabyte (PB) while the QLC SSD have estimated 450 TB of total write byte (TBW). Since QLC SSD is targeted at read-intensive workloads (e.g., 90%+ reads), it can endure up to 4.5PB data bytes that passed through the SSD interface within life-cycle, that is 28.5% higher than HDDs.
- **Cost-effective**: HDDs are still the most affordable storage solution in terms of the dollar per GB of storage (i.e., $ / GB). But in order to support the quality of service (QoS) demand that are essential to datacenter operation, a large number of HDD arrays is required to achieve the desired read performance. My previous study [71]

indicates that a RAID-5 array comprises of five HDDs or a RAID-6 array comprises of seven HDDs provide similar read performance to a single QLC SSD. Therefore, QLC SSD and HDD ended up having similar investment per GB to reach same level of performance. Consider the cost per GB for management and maintenance such as cooling and rack space, QLC SSD becomes more cost-effective than HDD, which leads to a higher investment gain.

CHAPTER 6

AN IN-LAB STUDY OF QLC SSD PERFORMANCE IN ACCELERATION TESTS

In this chapter, I would introduce the Highly Accelerated Life Tests (HALT) of NVMe QLC SSD. HALT is a reliable method to find design defects and weaknesses in electronic and electro-mechanical assemblies in the industry[4]. It includes a series of multi-stress tests that accelerate the aging of the SSD. By sequentially increases the stress of environments well beyond the SSD will encounter in normal usage, I simulate thermal tests from HALT in lab to evaluate the performance and reliability of QLC SSD.

6.1. Evaluating NVMe QLC SSD

This experiment focus on the NVMe QLC SSD. SATA and PCIe are the 2 major types of connectors in SSDs. SATA connectors are more seen on SLC, MLC, and TLC SSDs. Working with power connectors, SATA-based SSDs can have up to 6GB/s transportation speed. However, PCIe connectors are more popular in QLC SSDs. NVMe(Non-Volatile Memory Express), an interface protocol built specifically for SSDs, works with PCIe to transfer data to and from SSDs. SSDs with NVMe protocol(aka, NVMe SSDs) can provide up to 32GB/s transportation speed based on PCIe 3.0 without an additional power connector. Also, since the PCIe 4.0 already released, new SSDs adopt this protocol can be expected to twice as fast as PCIe 3.0, about 10x faster than SATA. Besides, the physical size of the NVMe SSD is normally smaller than SATA-based SSD. It also usually eliminates the outer protection box of each disk, and the user can see the NAND chip directly. Compared with other types of SSDs, QLC SSDs have higher capacities on the same size NAND chip. As I notice, many data center-level and enterprise-level SSDs are now NVMe SSDs.

In this experiments, I test 2 models of NVMe QLC SSDs. Even though both of them come from the same manufacturer, they are from different product series. Table 6.1 shows more detail information of them.

Table 6.1. Types of QLC SSD in Thermal Tests

Model*	**Cell Type**	**Architecture**	**Capacity**	**Firmware**	**Cost**
Model A	NVMe QLC	3D	512GB	004c	$
Model B	NVMe QLC	3D	118GB	k4110440	$$$

* For each model, the logical sector size are 512 byte; physical sector size may vary.

6.2. Experiment Setup

The HALT includes stress tests of temperature, rapid thermal cycle and vibration. In my experiment, I facility the thermal tests to QLC SSD. I monitor the QLC SSD behaviors during the process and analyze the performance and reliability degradation of the SSD. In the experiments, I utilize the Flexible I/O Tester (FIO) and HiBench to test my SSDs. In the first part of my experiments, I apply FIO to test some basic I/O performance of QLC SSDs, while in the second part, I apply HiBench to test big data benchmarks on SSD.

In the above mentioned experiments, I use the HP ML110 server with 16GB memory installed. The operating system is Ubuntu 20.04. All the packages, including FIO, HiBench, and other monitoring tools are installed in Ubuntu Environment.

6.2.1. FIO and Blktrace

FIO provides huge flexibility of defining workloads by users. There can be any number of processes or threads involved, and they can each be using their own way of generating I/O defined by users[3]. In my experiment, I unitize FIO in two different ways. The first usage is basic – to test the QLC SSD's throughput, IOPS and latency. This is included in the first part of my experiment. In the second part of the experiment, I cooperate the FIO with the Blktrace package, to replay the I/O traces in sufficient times. Blktrace is a Linux kernel based mechanism that provides detailed traces information of the I/O traffic on block devices[1]. User can see the kernel events and queue operations background details by utilizing Blktrace. Blktrace also can record those I/O information into a binary file and allow FIO to replay it in the future.

6.2.2. HiBench and Hadoop

HiBench and Hadoop are applied in the second part of the experiment. HiBench is a big data benchmark suit introduced by Intel. It is still actively update and the latest version is 7.1.1[7], which I use in the experiment. HiBench includes different types of big data workloads, including micro-benchmarks, machine learning benchmarks, web search benchmarks and database benchmarks. It supports these benchmarks running on Hadoop and Spark. In my experiment, I test some benchmarks on top of the Hadoop cluster. Despite that more benchmarks can be run on Spark, Hadoop is more capable to process big data sets that size exceeds available memory. Moreover, Hadoop is better used on batch process with tasks that exploit disk read and write operations[24]. Under these considerations, Hadoop is more suitable for my experiment purposes.

I set up the Hadoop cluster in standalone mode and enforce all the data read and write operations go through the QLC SSD testing subject. In this case, the HDFS, including the datanode and the namenode are assigned in the QLC SSD. Data sets for testing are also stored in it.

6.2.3. Thermal Environments

Thermal tests from HALT requires aging the test article, in this case, QLC SSD, through sequential steps to increase the stress environments. By following this rule, I set up the different temperature levels. But before that, I need to figure out the upper bound and lower bound of the temperature.

The upper bound temperature should be the thermal throttling throttling temperature of the SSD. As the temperature increase well beyond the thermal throttling temperature, the thermal throttling protection will be triggered. In extreme cases, the SSD will disconnect from the server, and cause the running process shut down immediately. The default thermal throttling threshold for NVMe drives are at 66°C. So, the temperature should not be set up beyond that. In addition, according to the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), it recommends that server inlet temperatures be between 18°C and 27°C (64.4°F to 80.6°F). For the up-time servers, however, it recommends an upper limit of 25°C[2].

Besides, considering there is not only storage media exists, other hardwares may also be installed in datacenters, Table 6.2 and 6.3 show the safe temperature of some other hardwares. The temperature values in the table relate to industrial-grade equipments. Comparing with the commercial-grade hardwares, the industrial-grade equipments can bare a higher range of temperature changes. Table 6.2 shows the idle temperature of the devices and the maximum temperature the devices may reach for intensive workloads. Table 6.3 shows the environmental temperature ranges that the devices can run continuously and safely. Most SSDs are rated for running within a temperature range of 0ºC up to a max temperature of 70ºC [23]. Specially, industrial-grade NVMe SSD can deliver stable performance even in extreme temperatures ranging from -40°C to 85°C [22]. However, please beware that the safe temperature does not indicate the devices can run at their best performance inside the ranges.

According to the above suggestions and considerations, I set up the temperature at 25°C, 35°C, 45°C and 50°C for QLC SSD running environment. According to the most datacenter temperature setup, I assume the QLC SSD running at the 25°C environment temperature is the normal usage case. I use the environment temperature at 25°C as the baseline. Then, I test the drive performance at 35°C, 45°C and 50°C. For 25°C, 35°C, and 50°C temperature setup, I test both the FIO and the Hadoop benchmarks. For all the temperature setups, I also test the Hadoop big data benchmark as well. I monitor the performance of the SSD drives during different temperature environments and different workloads.

Table 6.2. Safe Running Temperature of CPU, GPU and DRAM

Component	**CPU**[51]	**GPU**[47]	**DRAM**[98]
Idle Temperature	40-45°C	30-45°C	45°C
Intensive Loading Temperature	95°C	85°C	95°C

Table 6.3. Safe Running Temperature of Network and Storage

Component	**Network Card**[81]	**Network Switches**[11]	**HDD**[25]	**NVMeSSD**[22]
Lower-bound Temperature	0	-40°C	0°C	-40°C
Upper-bound Temperature	90°C	85°C	60°C	85°C

6.3. Performance Evaluation

6.3.1. FIO Benchmark Experiments

In this section, I test the QLC SSD under 25°C, 35°C, and 50°C using basic FIO command lines. I test their throughput, IOPS and latency. For all the categories, I test their sequential reading and writing, as well as random reading and writing. The following Table 6.4 and Table 6.5 show the results from Model A and Model B. Each test are repeated on 3 different SSD with the same model, so the results are average values.

From Table 6.4 and Table 6.5, both Model A and Model B QLC SSD shows similar behavior on performance and degradation when environment temperature increase from 25°C to 50°C. But still can see some differences between them.

6.3.1.1. Benchmark Results and Degradation Analysis

Table 6.4. Model A Throughput, IOPS and Latency Results

Benchmarks		**Throughput(MB/s)**			**IOPS**			**Latency(μs)**		
Temperature		*25°C*	*35°C*	*50°C*	*25°C*	*35°C*	*50°C*	*25°C*	*35°C*	*50°C*
Sequential	write zero	489.20	462.68	419.34	111154.06	61741.12	81527.78	39.19	44.20	47.46
	read zero	1339.71	1142.67	686.95	84253.43	64027.69	76975.22	18.04	20.60	19.35
	write random	518.89	455.17	442.80	119787.16	67122.74	90618.69	25.19	41.22	43.79
	read random	1288.10	1144.08	698.58	82767.31	63223.44	80812.20	18.11	20.99	19.20
Random	random read	1652.14	1316.74	387.29	84779.91	71160.59	42219.64	100.48	101.42	130.45
	random write	273.28	265.02	174.75	35946.24	28676.39	18579.51	34.61	40.43	95.95

Table 6.5. Model B Throughput, IOPS and Latency Results

Benchmarks		**Throughput(MB/s)**			**IOPS**			**Latency(μs)**		
Temperature		*25°C*	*35°C*	*50°C*	*25°C*	*35°C*	*50°C*	*25°C*	*35°C*	*50°C*
Sequential	write zero	594.57	580.60	425.38	135355.45	138732.57	108127.24	17.95	16.57	17.98
	read zero	1368.11	1302.76	1112.41	323729.82	332679.82	284557.43	12.04	10.95	12.12
	write random	594.38	580.40	434.41	135254.17	138623.00	109172.18	17.94	16.46	17.70
	read random	1368.32	1303.44	1078.78	322955.72	332516.35	275316.90	12.15	11.08	12.19
Random	random read	1368.12	1302.08	1083.03	324254.89	442449.81	275220.02	12.19	10.98	12.32
	random write	587.19	577.45	419.91	134667.87	138015.14	105696.33	17.69	16.52	17.72

For both Model A and Model B QLC SSD, when they are at 25°C, their throughput reach the maximum values as their manufacture advertised. For both models, reading throughput is much higher than writing throughput, no matter in sequence or random category. However, Model A seems more capable to achieve the best performance in sequential write than random write, but vise verse for reading operations. For Model B, writing and reading throughput are stable no matter it is sequential or random. For IOPS, Model B performs better than Model A, especially for reading operations. For Model A, it shows the writing operations have higher IOPS than reading operations for sequential read/write, but opposite for random read/write. For Model B, IOPS values are more consistent, no matter for sequential or random operations. And reading operations are having much higher values than writing operations. For Latency, reading operations perform much better than writing operations for both models, except the random read/write from Model A.

When temperature increase from 25°C to 35°C, the throughput of both models are downgraded. Overall, Model A shows larger degradation than Model B. Model A has the largest throughput degradation on random read, while Model B shows all the reading operations, including sequential reading and random reading, have similar degradation. Also, both models show their reading operations have a little higher degradation than their writing operations. Look at the IOPS values of Model A, IOPS values show significant degradation. However, the IOPS values of Model B show a little improve, especially the random read operation. From the Latency category, Model A shows its latency values increase when temperature increase from 25°C to 25°C. However, Model B shows its latency values decrease a little bit. In summary, when the temperature increase from 25°C to 35°C, Model A SSD shows degradation on throughput, IOPS and latency. However, Model B SSD only shows the degradation on I/O throughput. At the same, Model B shows a slightly improvement on IOPS and latency instead.

If the temperature increase from 25°C to 50°C, all the categories – throughput, IOPS and latency, show degradation on values. For throughput and IOPS, both Model A and Model B show obvious downgrading. Compared within 2 models, Model A has higher degradation than Model B, especially on reading operations. Please refer to later discussion in this section fro more details. The degradation of IOPS are similar between 2 models. For latency, Model A shows a clearly downgrading. Even though Model B also show downgrading, but the degradation is not that distincted as Model A. In order to have more insightful ideas of the degradation, I also calculate the downgrading percentages for both models. Table 6.6 and Table 6.7 show the percentage changes with temperature increasing from 25°C.

Also, here shows the equations of calculating the degradation percentages. For throughput and IOPS values, the larger the better, so I assume that the throughput and IOPS values in lower temperature would be larger than that in higher temperature. However, for the latency values, the smaller the better, so I assume the latency values in lower temperature would be smaller than that in higher temperature.

$$DegradationOfThroughput/IOPS = \frac{l-h}{l} \times 100\%$$

where,

l is the throughput/IOPS value in lower temperature,

h is the throughput/IOPS value in higher temperature.

Similar, the degradation percentage of latency equation shows below.

$$DegradationOfLatency = \frac{h-l}{l} \times 100\%$$

where,

l is the latency value in lower temperature,

h is the latency value in higher temperature.

Table 6.6. Model A Throughput, IOPS and Latency Degradation – 25°C

Benchmarks		**Throughput(MB/s)**		**IOPS**		**Latency(μs)**	
Temperature		*25°C → 35°C*	*25°C → 50°C*	*25°C → 35°C*	*25°C → 50°C*	*25°C → 35°C*	*25°C → 50°C*
Sequential	write zero	5.42%	14.28%	44.45%	26.65%	12.78%	21.10%
	read zero	14.71%	48.72%	24.01%	8.64%	14.21%	7.28%
	write random	12.28%	14.66%	43.97%	24.35%	63.64%	73.84%
	read random	11.18%	45.77%	23.61%	2.36%	15.88%	5.95%
Random	random read	20.40%	76.56%	16.06%	50.20%	0.93%	29.82%
	random write	3.02%	36.05%	20.22%	48.31%	16.80%	177.20%

Table 6.7. Model B Throughput, IOPS and Latency Degradation – 25°C

Benchmarks		**Throughput(MB/s)**		**IOPS**		**Latency(μs)**	
Temperature		*25°C → 35°C*	*25°C → 50°C*	*25°C → 35°C*	*25°C → 50°C*	*25°C → 35°C*	*25°C → 50°C*
Sequential	write zero	2.35%	28.46%	-2.50%	20.12%	-7.69%	0.17%
	read zero	4.78%	18.69%	-2.76%	12.10%	-9.05%	0.66%
	write random	2.35%	26.91%	-2.49%	19.28%	-8.25%	-1.34%
	read random	4.74%	21.16%	-2.96%	14.75%	-8.81%	0.33%
Random	random read	4.83%	20.84%	-36.45%	15.12%	-9.93%	1.07%
	random write	1.66%	28.49%	-2.49%	21.51%	-6.61%	0.17%

Both tables show obvious degradation happening on I/O bandwidths when temperature increases from 25°C to 50°. When temperature increases from 25°C to 35°C, the degradation percentages are small, while as the temperature keeps increasing, the percentages enlarge. Compared with Model A, Model B shows more stable degradation among reading and writing operations. For example, when temperature increases from 25°C to 50°C, all the degradation percentages are greater than 18% but lower than 30%. However, the degradation percentages of Model A are ranging from 14% to 76% under the same temperature changes. While comparing the degradation percentages between reading and writing, Model A shows the reading operations have a larger performance dropping than the writing, especially for the random read. But the Model B has different results. Model B shows only its reading operation degradation percentages greater than its writing's when temperature from 25°C to 35°C. But it shows its writing operations sensitive to temperature increases than its reading's when temperature increases from 25°C to 50°C.

For IOPS, the degradation percentages of writing operations are higher than the readings' in Model A. And for the sequential I/O operations, the majority degradation happens when temperature increases from 25°C to 35°C. For Model B, when the temperature reaches 35°C, the IOPS performs a little better than that in 25°C, especially the random read operations. However, it is really hard to tell whether those percentage changes are real or just statistic errors since the percentages are similar and small, which are close to 2.5%. To continue increasing the temperature to reach 50°C, the degradation happens.

For latency, Model A shows its writing latency are larger than its reading's. The latency of Model B has a little improvement while temperature increases to 35°C. But later the latency seems back to normal when temperature reaches 50°C.

Overall, Model A shows higher percentages of degradation on throughput, IOPS and latency, which indicates that Model A SSD is sensitive and reactive to higher temperature environments, especially for the reading operations. Compared with Model A, Model B shows its stable and consistent degradation percentages among all the sequential I/O and random I/O.

6.3.1.2. P-values and Analysis

Comparing the performance results on 25°C, SSD shows significant performance downgrade on values and percentage when temperature reaching at 50°C. To prove this conclusion in more confidence, I also facilitate the T-test to evaluate the significant level using p-values to demonstrate the degradation of each benchmarks.

T-test is a type of inferential statistic to determine if there is a significant difference between the means of two groups. In order to analyze the significant intervals, I use two-tail equal variance T-test. The P-values in my experiment are calculated by the following formula.

$$P - value = \frac{mean1 - mean2}{\frac{(n1-1)\times var1^2+(n2-1)\times var2^2}{n1+n2-2} \times \sqrt{\frac{1}{n1} + \frac{1}{n2}}}$$

where,

mean1 and *mean2* are average values of each sample sets,

var1 and *var2* are variances,

n1 and *n2* are number of records.

P-values are probability values to describe how likely it is the data would be occurred randomly [28]. The value of P-value ranges from 0 to 1. The smaller the p-value, the stronger the evidence that we should reject the null hypothesis. Thus, in my cases, the **null hypothesis** is **QLC SSD shows no performance different between 25°C and 50°C**. Normally, a p-value less than 0.05 is statistically significant. It indicates there are less than 5% probability that the null hypothesis is acceptable.Similarity, a p-value less than 0.01 is statistically extremely significant. It indicates that there are less than 1% probability that the null hypothesis is acceptable. Table 6.8 shows the p-values of the Model A SSD of different benchmarks, and Table 6.9 shows Model B's p-values.

The majority of p-values from both models show significant degradation, some benchmarks' p-values even show extremely significant. For Model A, most benchmarks in throughput show their p-values less than 0.01, except the write zero benchmark shows no significant at all. In IOPS, the read zero benchmark shows no significant while others are all show significant. In Latency, only random read benchmark and random write benchmark show significant degradation in Model A. Compared with the sequential I/O operations, the random I/O operations are more likely to have degradation due to temperature increase. For Model B drive, it shows extremely significant degradation for all the benchmarks in throughput and IOPS categories, but no significant in the Latency category.

After analyzing the FIO benchmark results in average values, percentages and p-values, I can confirm that **Finding 1: QLC SSD show degradation in throughput and IOPS when temperature increases beyond the recommended their normal operating temperature.** But the experiment also confirm that **SSD Latency is rarely affected by the temperature increases.**

Moreover, when comparing the two different SSD models, I also find out that even though both models show throughput degradation, but different models reacts differently. Model A shows more sensitive and acclimate to temperature increases when doing reading operations. However, Model B shows its sensitive to writing operations more than reading operations in high temperature. Its writing downgrading percentages are slightly higher than its readings' at 50°C. **Finding 2: Different QLC SSD model show different degradation preference on read/write.**

Furthermore, when comparing the overall performance between two models, I also can see that Model B shows a higher ability to maintain stable and balance on throughput and, IOPS and Latency even after degradation. For example, when temperature increases from 25°C to 50°C, the throughput degradation percentages of Model A can be vary from 14% to 76%, but Model B's ranges from 15% to 28%. Similar scenario I observed from IOPS. Without further investigation, I do not know the exact cause of it. But I believe that the their different firmware is part of the cause.

6.3.2. Big Data Benchmarks Experiments

The FIO benchmark result provides an overall basic idea of how those QLC SSD models react to temperature increases. However, real-world application scenarios are more complicated. Especially considering the big data booming recently, as well as the QLC SSD is targeting the high-end storage market, analyzing more practical cases may help to have a deeper understanding of the QLC SSD's reliability when facing temperature increases. In this section, I test the QLC SSD Model A and Model B under 25°C, 35°C, 45°C and 50°C environment temperatures. I apply different big data workloads on top of them, and observe their degradation behaviours on I/O throughput.

Table 6.8. Model A P-values and Significant

Benchmarks		**Throughput(MB/s)**	**IOPS**	**Latency(μs)**
Temperature		*25°C vs.50°C*	*25°C vs.50°C*	*25°C vs.50°C*
Sequential	write zero	1.91E-01$^{\mathrm{NS}}$	1.78E-02 *	4.49E-01$^{\mathrm{NS}}$
	read zero	9.95E-05 **	8.80E-02$^{\mathrm{NS}}$	9.59E-02$^{\mathrm{NS}}$
	write random	8.51E-04 **	1.16E-03 **	1.32E-01$^{\mathrm{NS}}$
	read random	7.26E-04 **	4.76E-02 **	1.68E-01$^{\mathrm{NS}}$
Random	random read	1.79E-08 **	5.94E-07 **	1.09E-04 **
	random write	3.70E-03 **	1.13E-02 *	3.81E-02 *

$^{\mathrm{NS}}$ indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value ≦ 0.05, which the performance different is significant;

** indicates p-value ≦ 0.01, the different is extremely significant.

Table 6.9. Model B P-values and Significant

Benchmarks		**Throughput(MB/s)**	**IOPS**	**Latency(μs)**
Temperature		*25°C vs.50°C*	*25°C vs.50°C*	*25°C vs.50°C*
Sequential	write zero	8.85E-05 **	1.17E-04 **	8.39E-01$^{\mathrm{NS}}$
	read zero	7.41E-04 **	4.52E-03 **	4.01E-01$^{\mathrm{NS}}$
	write random	3.51E-05 **	2.81E-04 **	5.56E-01$^{\mathrm{NS}}$
	read random	9.12E-04 **	2.80E-03 **	2.49E-01$^{\mathrm{NS}}$
Random	random read	9.59E-04 **	1.94E-03 **	8.49E-01$^{\mathrm{NS}}$
	random write	1.01E-04 **	2.04E-04 **	2.10E-01$^{\mathrm{NS}}$

$^{\mathrm{NS}}$ indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value ≦ 0.05, which the performance different is significant;

** indicates p-value ≦ 0.01, the different is extremely significant.

The experiment has two steps. First of all, I set up the Hadoop environment and run each benchmark on top of it under 25°C. During the same time, I integrate the Blktrace to record the I/O traces from the process. During this step, the I/O traces of Model A and Model B SSD will be recorded separately. Secondly, I apply the FIO to replay the recorded traces 100 times under different temperature environments and collect their throughput data as well as data from other monitoring tools. The repetition time is defined by my pre-experiment. The QLC SSD would not show degradation from the first few times of replay. And different benchmark show different start time of degradation. The 100 is a number large enough that throughput of all benchmarks show degradation, and is stable for a long enough time without showing second degradation[1].

6.3.2.1. Data Size

Due to the server memory limitation, I use different size of data sets for different benchmarks in order to approach the maximum usage of the memory. Table 6.10 summarizes the data set sizes of each workload. Data set sizes are pre-defined by HiBench.

6.3.2.2. Benchmarks and I/O Trace Analysis

The workloads in this experiment section include micro-benchmarks, machine learning benchmarks and web search benchmarks. From the micro-benchmarks, I test the Wordcount, Sort, Dfsioe-read and Dfsioe-write benchmarks. From the machine learning benchmarks, I test the Bayes and Kmeans benchmarks. From the web search benchmarks, I apply the Nutchindexing and Pagerank benchmarks.

- **Wordcount:** The Wordcount benchmark is one of the simple applications contained in the Hadoop distribution. It takes a set of text files as input, and counts the number of times that each word appears[26]. While running the Wordcount benchmark on Hadoop standalone mode, the whole process includes tow parts: Map and Reduce. The Mapping process takes over 90% of the time process time. And the Reduce process run after the Mapping process finishing.

[1] Second degradation is not observed even repetition reach higher number.

Table 6.10. Benchmarks and Data Set Sizes

Benchmarks		**Dataset**	**Size**
Micro	Wordcount	Huge	32,000 MB
	Sort	Huge	3,200 MB
	Dfsioe-read	Huge	256 × 100 MB
	Dfsioe-write	Huge	256 × 100 MB
Machine Learning	Bayes	Large	pages: 100,000 classes: 100 ngrams: 2
	Kmeans	Large	number of clusters: 5 dimensions: 20 number of samples: 20,000,000 samples per input file: 4,000,000 maximum iteration: 5 k: 10
Web Search	Nutchindexing	Small	pages: 1,000,000
	Pagerank	Small	pages: 5,000 number of iterations: 3 block: 0 block width: 16

- **Sort:** The Sort benchmark sorts a set of records that is randomly generated. The application uses identity map and identity reduce functions as the MapReduce framework does the sorting[26].
- **Dfsioe:** The Dfsioe benchmark in the HiBench suit is the enhanced DFSIO. It tests the HDFS throughput of the Hadoop cluster by generating a large number of tasks performing writes and reads. It measures the average I/O rate of each map task, the average throughput of each map task, and the aggregated throughput of HDFS

cluster[7].

- **Byes:** The Bayes is a simple classical classification machine learning algorithm. In HiBench, it generate documents whose words follow the Zipfian distribution[7]. During the process, Map and Reduce run simultaneously.
- **Kmeans:** The Kmeans workload is a well-known clustering machine learning algorithm. In HiBench, the Kmeans benchmarks generates input from GenKMeans-Dataset based on Uniform Distribution and Gaussian Distribution[7]. In my experiment, the default cluster number is five. While running the Kmeans workload, MapReduce jobs run cluster by cluster until all cluster are finished.
- **Nutchindexing:** The Nutchindexing workload tests the indexing sub-system in Nutch. It uses the automatically generated Web data whose hyperlinks and words both follow the Zipfian distribution with corresponding parameters[7].
- **Pagerank:** The Pagerank is a search engine ranking algorithm. The Pagerank benchmark test the Hadoop embedded pagerank algorithm using the data source generated from Web data whose hyperlinks follow the Zipfian distribution[7].

Even though all the HiBench benchmarks are running MapReduce on top of Hadoop, different benchmark has unique workload patterns, as well as different ratios of read and write. Table 6.11 shows the read/write ratios for each benchmark from both SSD models. Since the Pagerank benchmark does not have any reading entrance, so the table only shows the raw number of reading and write entrance.

$$Ratio(read/write) = \frac{n1}{n2}$$

where,

n1 = number of read entrance,

n2 = number of write entrance.

From the table, the Wordcount and the Nutchindexing benchmark show reading operation is more intensive than writing operations. Especially the Wordcount benchmark shows its reading entrances are over double than the its writings'. But other benchmarks are write entrances over read entrances, even the Dfsioe-read benchmark. However, these ratios only show the frequency of reading and writing operations, but do not indicate the I/O bandwidth. And also cannot define the workload is read intensive or write intensive at all. Cause each read/write entrance can have different I/O bandwidth.

Table 6.11. Benchmarks Read/Write Ratios

Benchmarks		**Model A**	**Model B**
Micro	Wordcount	2.05	2.37
	Sort	0.38	0.57
	Dfsioe-read	0.54	0.81
	Dfsioe-write	0.48	0.55
Machine Learning	Bayes	0.17	0.13
	Kmeans	0.62	0.69
Web Search	Nutchindexing	1.12	1.17
	Pagerank(raw)	0/90	0/98

In order to have a better understanding of the workload patterns, Figure 6.1 and Figure 6.2 show the I/O request numbers from each model.

In general, Model A and Model B show similar workload patterns in each benchmark. For the Wordcount benchmark, read and write requests distribute almost evenly after the the process starting. And the read requests are overwhelming the write requests during the whole process. So, the Wordcount benchmark is a read intensive benchmark. For the Sort benchmark, write request peaks after the middle of the process, and the request number is high to reach 7000 for Model A and 5000 for Model B. While the read request remains a low number through the whole process. Thus, The Sort benchmark is no doubt a write intensive benchmark. The Dfsioe-read and the Dfsioe-write benchmarks, as their names indicated, read request domains the Dfsioe-read, and write request domains the Dfsioe-write. Even though both of them are not purely read or write requests only. The read requests from the Bayes and the Kmeans benchmarks are only appear at some moment of the processes.

Specially, the write request peaks at the end of the process of the *Kmeans* benchmark. Combined with the read/write ratio from Table 6.11, the Bayes benchmark and the Kmeans benchmark are write intensive benchmarks. The *Pagerank* benchmark is a write intensive workload since it only has write operations. The *Nutchindexing* benchmark workload pattern shows four peaks. The first three peaks have both write and read requests. The 4th peak only shows write requests. Considering the read/write ratio of the benchmark, the *Nutchindexing* benchmark seems a read/write balance workload.

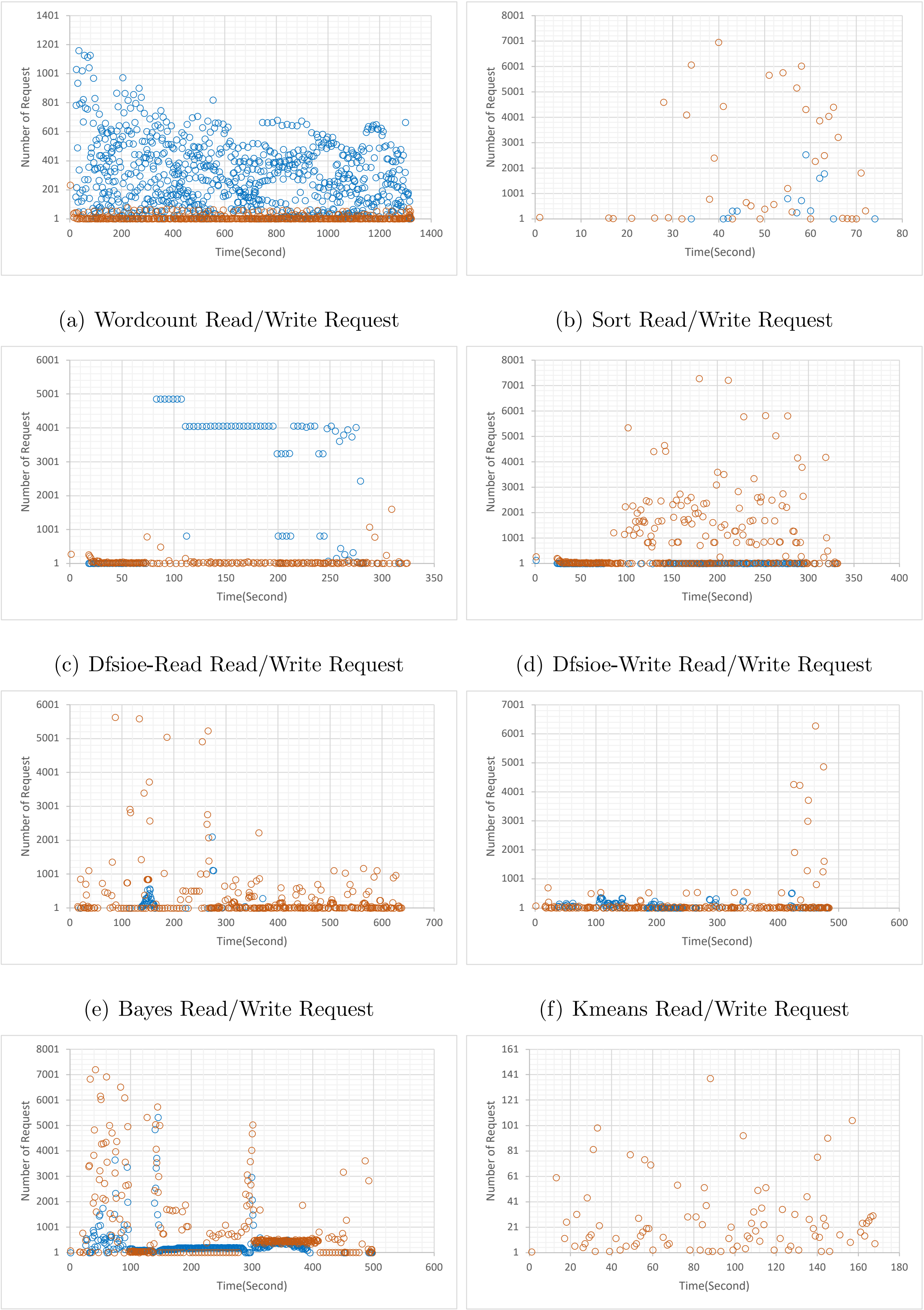
6.3.2.3. Benchmark Results and Degradation Analysis

Table 6.12 and Table 6.13 display the average values of I/O throughput from Model A and Model B in different environment temperature.

When temperature is less than and equal to 45°C, both Model A and Model B show their maximum reading throughput from the Wordcount benchmark. The read intensive benchmark – Dfsioe-read also show its throughput value close to the maximum. From the Dfsioe-write benchmark from both models show their maximum writing throughput. Other write intensive benchmarks, for example, the Sort, the Bayes and the Kmeans show relative high values on write. But the Pagerank benchmark shows lower I/O throughput values comparing with other write intensive benchmarks.

TABLE 6.12. Model A Throughput under Different Temperatures

	Temperature	**25°C**		**35°C**		**45°C**		**50°C**	
	Throughput(MB/s)	Read	Write	Read	Write	Read	Write	Read	Write
Micro	Wordcount	1411.26	5.21	1426.44	5.26	1443.36	5.33	1258.60	4.64
	Sort	82.32	777.51	84.57	798.68	82.96	783.61	75.118	710.82
	Dfsioe-read	1325.00	39.89	1333.22	40.15	1346.66	40.57	1211.80	36.48
	Dfsioe-write	0.09	878.88	0.09	879.34	0.09	842.89	0.08	766.32
Machine Learning	Bayes	73.68	710.12	79.15	763.13	78.19	754.11	65.74	633.71
	Kmeans	118.85	758.67	120.49	768.89	117.44	749.81	101.64	649.03
Web Search	Nutchindexing	223.07	534.30	221.57	530.71	218.84	524.31	181.62	435.12
	Pagerank	NA	167.45	NA	167.52	NA	165.48	NA	155.26



(a) Wordcount Read/Write Request

(b) Sort Read/Write Request

(c) Dfsioe-Read Read/Write Request

(d) Dfsioe-Write Read/Write Request

(e) Bayes Read/Write Request

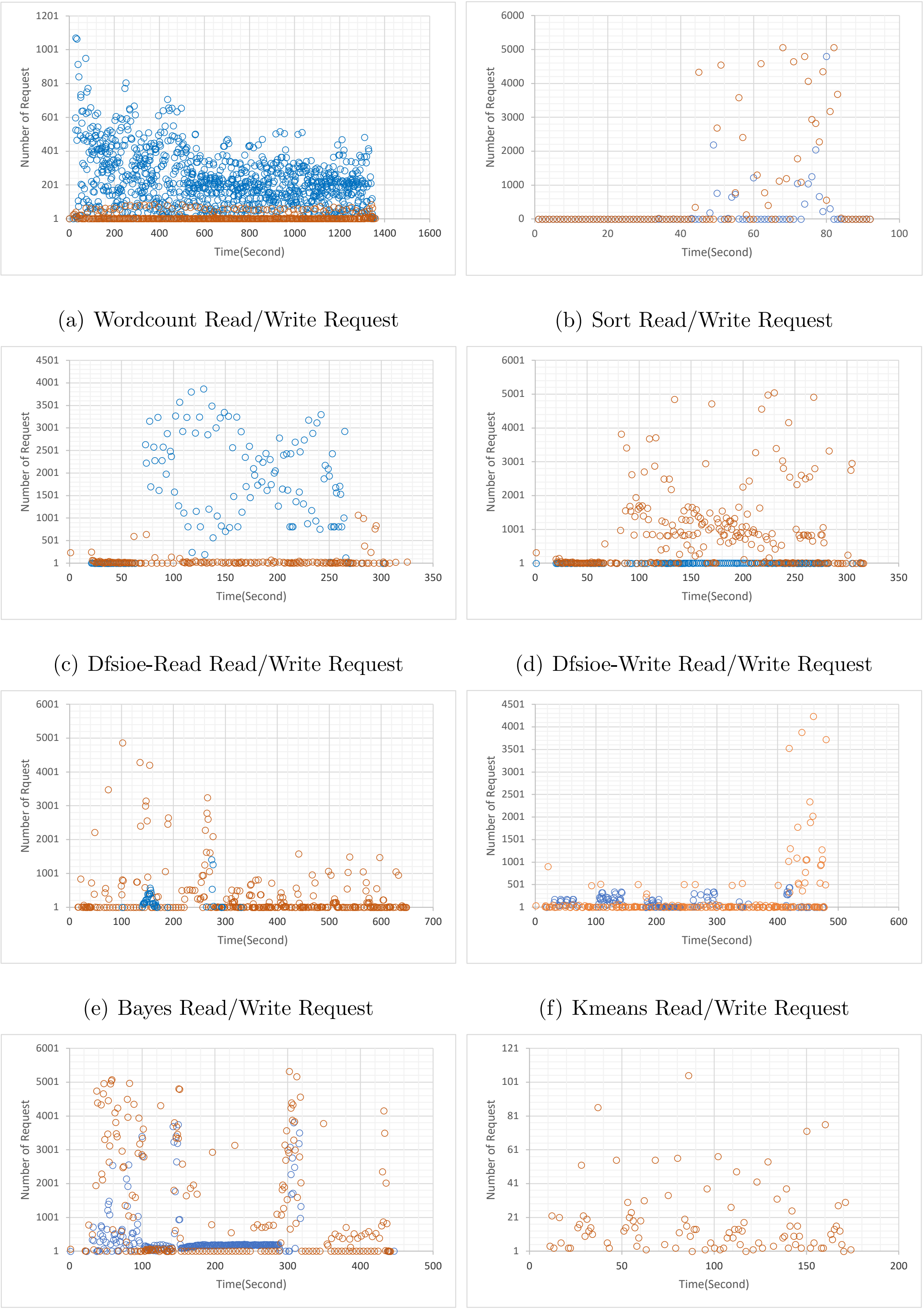
(f) Kmeans Read/Write Request

(g) Nutchindexing Read/Write Request

(h) Pagerank Read/Write Request

FIGURE 6.1. Summary of Read and Write Request from Model A

(the ○ indicates the Read, the ○ indicate the Write)



(a) Wordcount Read/Write Request

(b) Sort Read/Write Request

(c) Dfsioe-Read Read/Write Request

(d) Dfsioe-Write Read/Write Request

(e) Bayes Read/Write Request

(f) Kmeans Read/Write Request

(g) Nutchindexing Read/Write Request

(h) Pagerank Read/Write Request

Figure 6.2. Summary of Read and Write Request from Model B

(the ○ indicates the Read, the ○ indicate the Write)

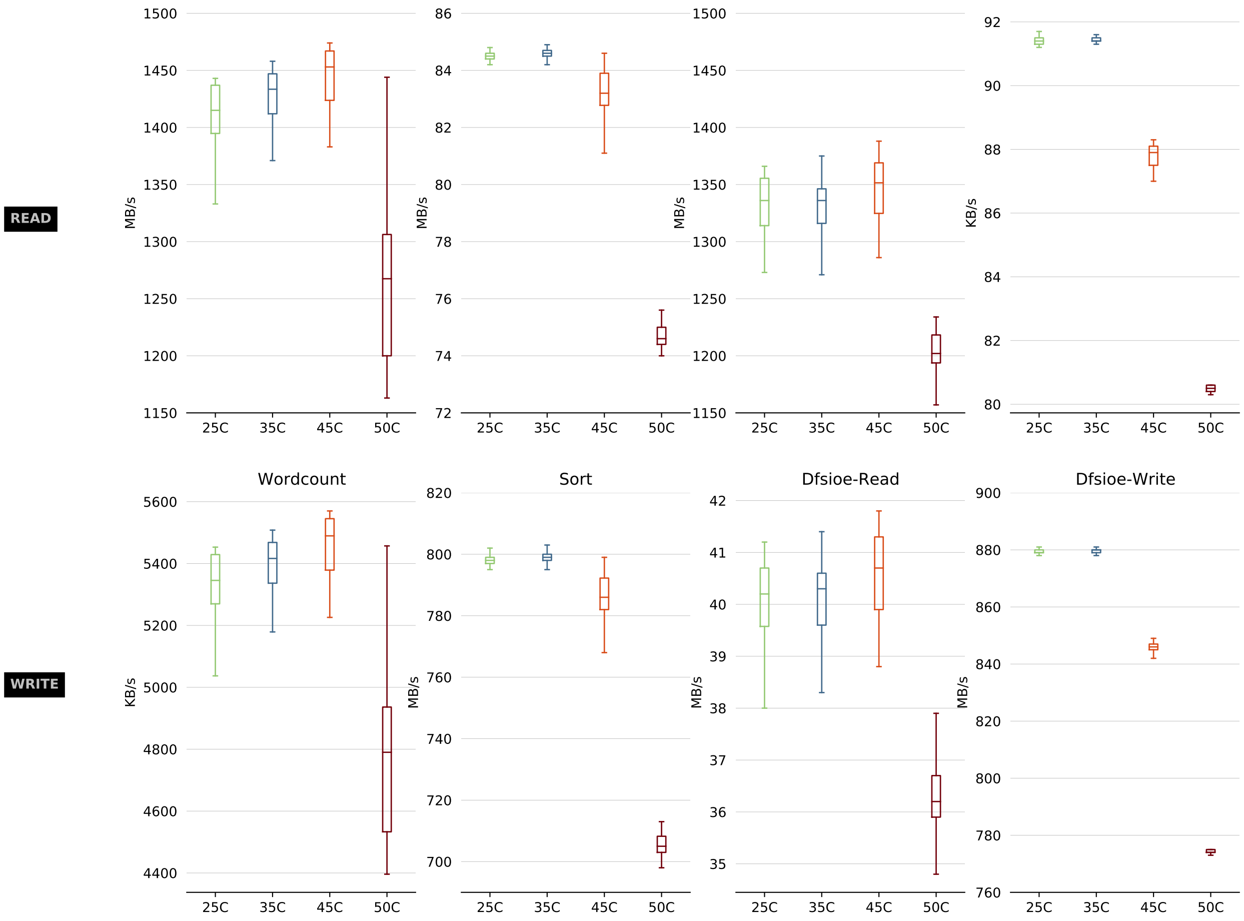


FIGURE 6.3. Distribution of Micro-Benchmarks from Model A

TABLE 6.13. Model B Throughput under Different Temperatures

	Temperature	**25°C**		**35°C**		**45°C**		**50°C**	
Throughput(MB/s)	Read	Write	Read	Write	Read	Write	Read	Write	
Micro	Wordcount	1251.92	4.64	1248.90	4.63	1242.98	4.60	931.92	3.45
	Sort	127.99	530.67	127.83	529.74	127.40	527.88	85.49	354.63
	Dfsioe-read	1179.52	38.97	1174.62	38.78	1173.83	38.78	898.39	29.67
	Dfsioe-write	0.06	586.81	0.06	588.60	0.06	576.44	0.04	357.49
Machine Learning	Bayes	50.61	562.66	50.55	564.17	51.09	565.51	32.85	364.28
	Kmeans	95.80	537.14	96.17	539.01	96.03	538.17	65.69	368.47
Web Search	Nutchindexing	198.95	490.39	199.70	491.57	199.51	491.19	127.20	313.28
	Pagerank	NA	437.59	NA	441.89	NA	444.79	NA	444.46

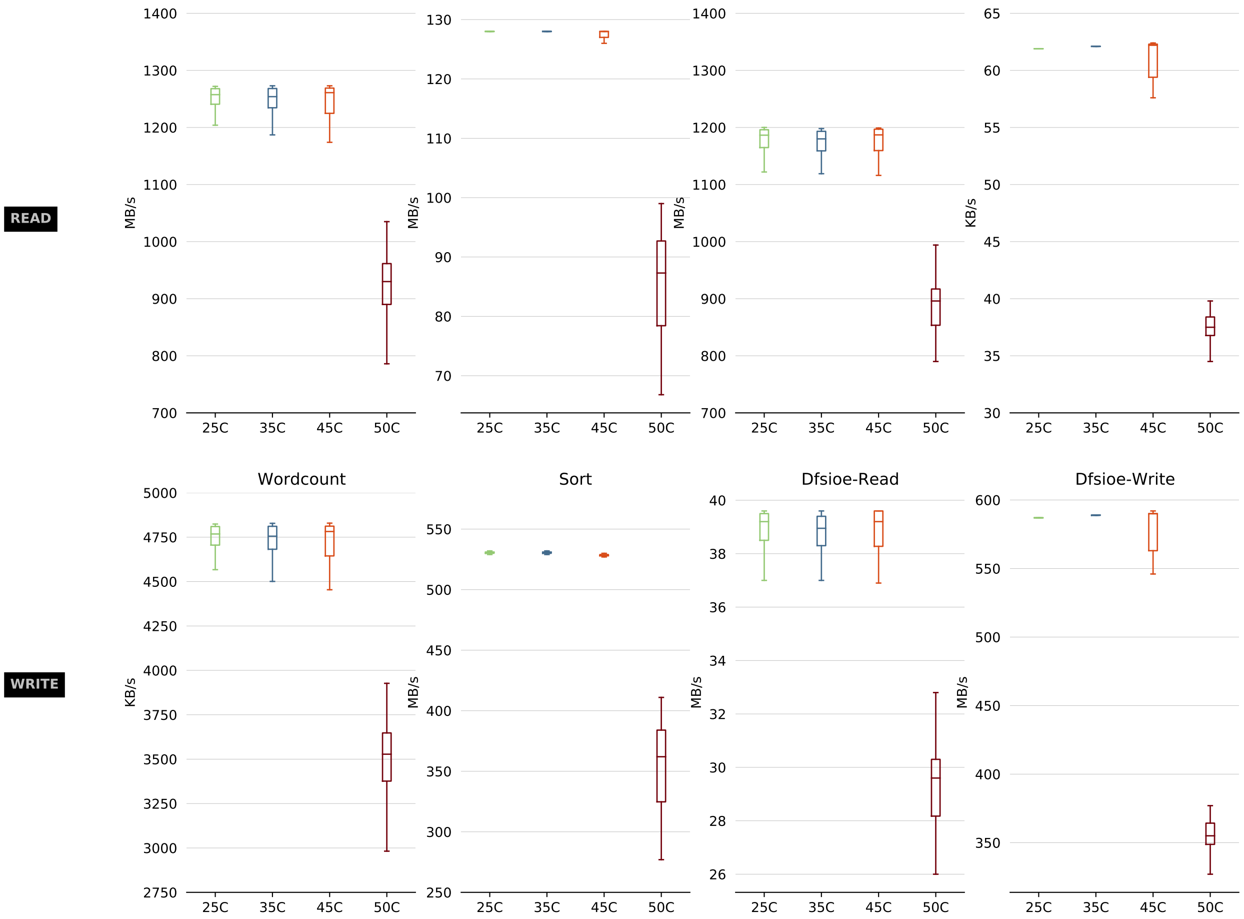
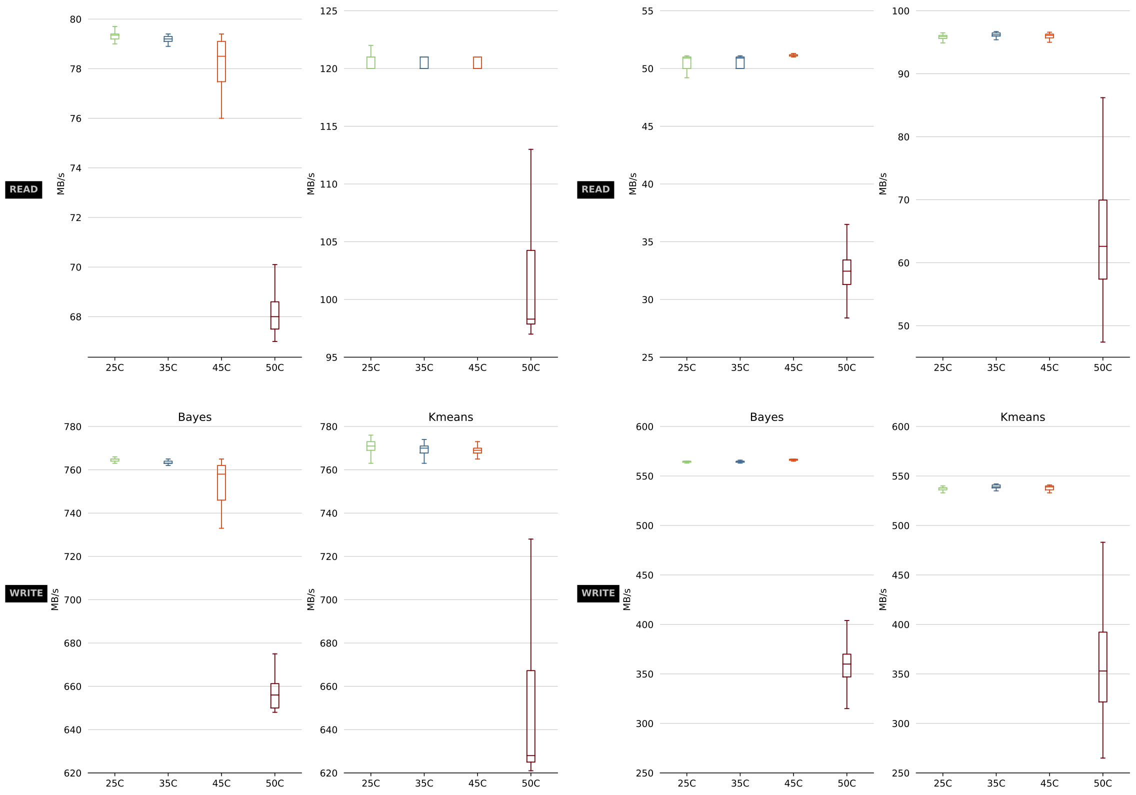


Figure 6.4. Distribution of Micro-Benchmarks from Model B

To have a deeper understanding, Figure 6.3, Figure 6.4, Figure 6.5 and Figure 6.6 show the distributions of I/O throughput of benchmarks using the box-plot.

Figure 6.3 and Figure 6.4 show the I/O distributions of Micro-Benchmark under different temperature. All benchmarks show significant I/O throughput dropping when temperature reach up to 50°C. For Model A, the throughput of the Wordcount benchmark and the Dfsioe-read benchmark, which are reading intensive workloads, even have a little increase when temperature increase from 25°C to 45°C. The sort and the Dfsioe-write, which are writing intensive workloads, their throughput start decreasing earlier at 45°C. For Model B, all the micro-benchmark show their throughput downgrading starting at 50°C. Before reaching the 50°C, their throughput are stable, not matter for reading intensive or writing intensive workloads.



(a) Distribution of Machine Learning Benchmarks from Model A

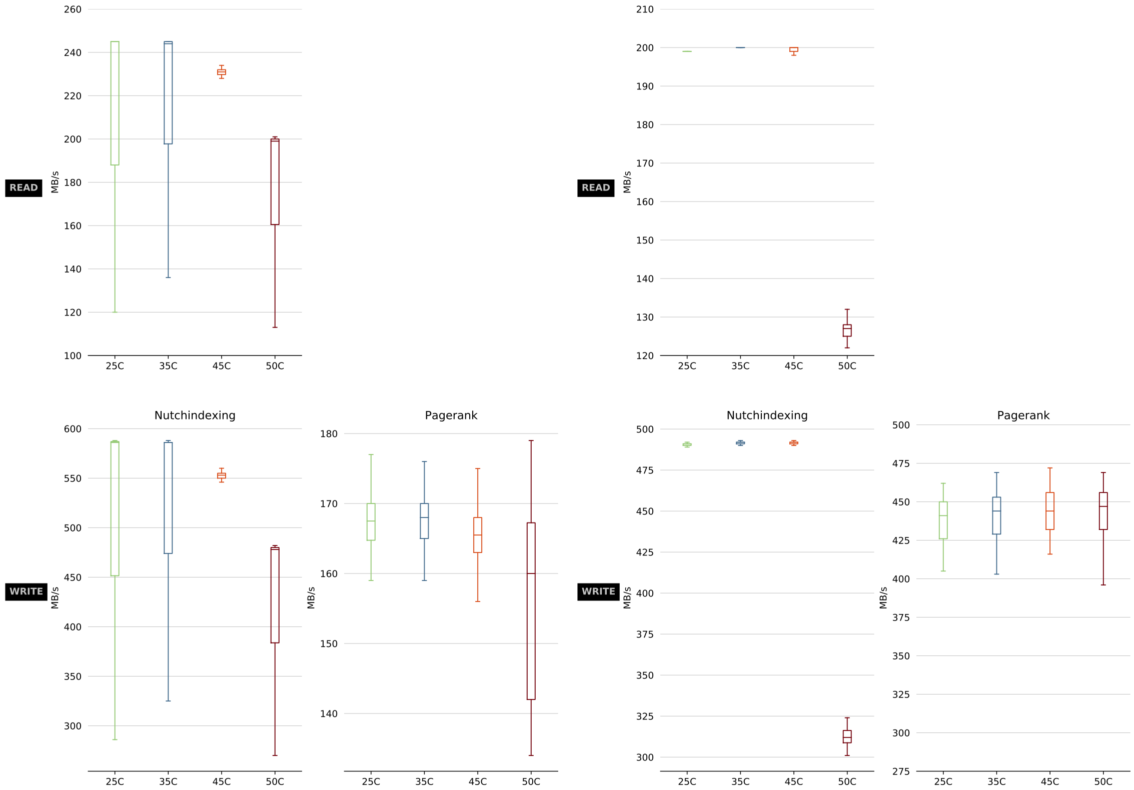
(b) Distribution of Machine Learning Benchmarks from Model B

Figure 6.5. Distribution of Machine Learning Benchmarks

Figure 6.5 shows the I/O distributions of Machine Learning benchmarks from both models. Both of them show major downgrade of I/O throughput at 50°C. The Bayes benchmark of Model A, however, its degradation starts at 45°C.

Figure 6.6 shows the I/O distribution of Web Search benchmarks from both models. Model A shows the major downgrading starting at 50°C. Model B only show the I/O throughput of the Nutchindexing benchmark downgraded at 50°C. However, the throughput of the Pagerank benchmark are close not matter at what temperature tier.

In a nutshell, this experiment reinforces my first finding: when the environment temperature increases well beyond the recommended operating temperature, QLC SSD show significant degradation in throughput. Furthermore, this experiment also confirm that **Finding 3: 50°C is the borderline temperature that major degradation starts**. Specially, Model A shows its throughput downgrading earlier at 45°C when running write intensive workloads. In the opposite, read intensive workloads shows a small throughput increases before 45°C. Besides, comparing Model B with Model A, Model B shows relatively consistent throughput on all benchmarks, no matter for read intensive or write intensive workloads. In addition, the box-plots at 50°C show larger boxes than at lower temperatures. It indicates that **Finding 4 : the fluctuation of throughput at 50°C is much higher than lower temperature.**



(a) Distribution of Web Search Benchmarks from Model A

(b) Distribution of Web Search Benchmarks from Model B

FIGURE 6.6. Distribution of Web Search Benchmarks

6.3.2.4. P-values and Analysis

Although the throughput values from the experiment show the major throughput degradation happens at 50°C, but they can not prove that there exist significant difference between throughput values at different temperature.

Table 6.14. Model A P-Values of Different Temperature Changes (25°C → 50°C)

	Temperature	**25°C** vs. **35°C**		**25°C** vs. **45°C**		**25°C** vs. **50°C**	
	P-Values	Read	Write	Read	Write	Read	Write
Micro	Wordcount	5.20E-05**	5.09E-05**	4.44E-15**	4.12E-15**	3.90E-51**	4.10E-51**
	Sort	3.94E-02*	3.81E-02*	5.57E-01^{NS}	5.51E-01^{NS}	8.52E-10**	8.73E-10**
	Dfsioe-read	3.04E-01^{NS}	2.77E-01^{NS}	8.04E-03**	5.98E-03**	3.17E-29**	4.14E-29**
	Dfsioe-write	6.74E-01^{NS}	4.22E-01^{NS}	1.13E-20**	1.88E-19**	2.03E-58**	2.15E-58**
Machine Learning	Bayes	1.07E-03**	9.92E-04**	6.82E-03**	6.15E-03**	5.55E-05**	5.71E-05**
	Kmeans	2.21E-01^{NS}	2.32E-01^{NS}	5.13E-01^{NS}	5.19E-01^{NS}	4.23E-26**	4.37E-26**
Web Search	Nutchindexing	7.91E-01^{NS}	7.91E-01^{NS}	3.77E-01^{NS}	3.84E-01^{NS}	7.53E-16**	7.78E-16**
	Pagerank	NA	9.08E-01	NA	3.25E-03**	NA	4.47E-15**

^{NS} indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value $\leqq 0.05$, which the performance different is significant;

** indicates p-value $\leqq 0.01$, the different is extremely significant.

Table 6.14 through Table 6.17 show the P-values of Model A and Model B for each benchmarks at different temperature. The majority of the p-values shows that the throughput values between different temperature having significant differences. From Table 6.14 and Table 6.14, as temperature increases from 25°C to 35°C, 3/8 benchmarks' p-values show significant. When the temperature increases from 35°C to 45°C, 3/4 benchmarks' p-values show extremely significant. As the temperature reaches 50°C, all benchmark's p-values show extremely significant. P-values of Model B show a similar trend. Taking the box-plot results into consideration, as the temperature keep increasing, more and more benchmarks show significant throughput degradation. When comparing p-values between different temperature, most p-values show minimum values at 50°C. Also, as the temperature increases, the p-values are smaller. This illustrates the degradation is more significant at higher temperature. Here, I summary the **Finding 5: When temperature beyond 25°C, QLC SSD throughput starts degradation for some benchmarks. As the temperature keep increasing, more and more benchmark workloads starting to degraded, and the degradation becomes more and more significant.**

Table 6.15. Model A P-Values of Different Temperature Changes (35°C → 50°C)

	Temperature	35°C vs. 45°C		35°C vs. 50°C		45°C vs. 50°C	
	P-Values	Read	Write	Read	Write	Read	Write
Micro	Wordcount	8.11E-06 **	8.09E-06 **	3.20E-57 **	3.38E-57 **	6.77E-63 **	6.94E-63 **
	Sort	6.04E-27 **	2.74E-27 **	1.51E-99 **	5.98E-100 **	8.56E-76 **	3.84E-76 **
	Dfsioe-read	4.30E-04 **	4.62E-04 **	1.07E-66 **	2.95E-65 **	4.07E-71 **	1.97E-70 **
	Dfsioe-write	6.71E-21 **	7.56E-20 **	1.39E-58 **	1.13E-58 **	6.78E-27 **	1.33E-27 **
Machine Learning	Bayes	9.54E-17 **	2.10E-17 **	2.16E-29 **	1.76E-29 **	2.56E-26 **	1.82E-26 **
	Kmeans	7.39E-02^{NS}	7.82E-02^{NS}	3.27E-102 **	2.70E-103 **	1.80E-16 **	1.22E-16 **
Web Search	Nutchindexing	6.04E-01^{NS}	6.13E-01^{NS}	8.05E-13 **	8.41E-13 **	1.31E-15 **	1.24E-15 **
	Pagerank	NA	1.07E-03 **	NA	1.47E-15 **	NA	2.15E-11 **

^{NS} indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value ≦ 0.05, which the performance different is significant;

** indicates p-value ≦ 0.01, the different is extremely significant.

Table 6.16. Model B P-Values of Different Temperature Changes (25°C → 50°C)

	Temperature	25°C vs. 35°C		25°C vs. 45°C		25°C vs. 50°C	
	P-Values	Read	Write	Read	Write	Read	Write
Micro	Wordcount	2.92E-01^{NS}	2.88E-01^{NS}	1.38E-02 *	1.35E-02 *	2.04E-107 **	1.85E-107 **
	Sort	4.64E-02 *	1.26E-02 *	8.67E-10 **	6.04E-14 **	6.21E-117 **	3.64E-117 **
	Dfsioe-read	8.05E-02^{NS}	5.45E-02^{NS}	9.46E-02^{NS}	1.06E-01^{NS}	6.23E-91 **	3.99E-90 **
	Dfsioe-write	6.49E-13 **	1.62E-12 **	6.03E-08 **	1.18E-09 **	7.73E-168 **	1.70E-169 **
Machine Learning	Bayes	4.58E-01^{NS}	1.86E-02 *	5.88E-10 **	1.13E-04 **	1.03E-120 **	1.46E-122 **
	Kmeans	2.86E-09 **	5.80E-10 **	4.39E-05 **	2.14E-04 **	9.72E-62 **	9.50E-62 **
Web Search	Nutchindexing	3.67E-13 **	4.12E-07 **	8.22E-07 **	2.57E-03 **	4.66E-200 **	9.18E-200 **
	Pagerank	NA	2.73E-02 *	NA	1.65E-04 **	NA	3.77E-04 **

^{NS} indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value ≦ 0.05, which the performance different is significant;

** indicates p-value ≦ 0.01, the different is extremely significant.

TABLE 6.17. Model B P-Values of Different Temperature Changes (35°C → 50°C)

	Temperature	**35°C** vs. **45°C**		**35°C** vs. **50°C**		**45°C** vs. **50°C**	
	P-Values	Read	Write	Read	Write	Read	Write
Micro	Wordcount	1.19E-01^{NS}	1.19E-01^{NS}	1.35E-105 **	1.29E-105 **	1.40E-100 **	1.32E-100 **
	Sort	4.58E-04 **	2.00E-04 **	2.88E-116 **	2.55E-116 **	2.47E-115 **	1.59E-115 **
	Dfsioe-read	8.19E-01^{NS}	9.66E-01^{NS}	2.45E-89 **	5.29E-88 **	7.88E-87 **	3.41E-86 **
	Dfsioe-write	2.56E-12 **	2.38E-12 **	1.94E-169 **	4.29E-170 **	5.46E-149 **	1.73E-149 **
Machine Learning	Bayes	4.38E-17 **	2.95E-04 **	3.60E-121 **	1.97E-124 **	4.72E-124 **	1.38E-124 **
	Kmeans	1.85E-02 *	5.89E-03 **	1.59E-62 **	1.88E-62 **	3.09E-62 **	3.88E-62 **
Web Search	Nutchindexing	1.24E-01^{NS}	1.64E-01^{NS}	1.89E-200 **	3.50E-200 **	8.95E-200 **	1.56E-199 **
	Pagerank	NA	1.37E-01^{NS}	NA	1.93E-01^{NS}	NA	8.63E-01^{NS}

^{NS} indicates p-value > 0.05, which the performance different is NOT significant;

* indicates p-value $\leqq 0.05$, which the performance different is significant;

** indicates p-value $\leqq 0.01$, the different is extremely significant.

CHAPTER 7

AVOIDING CATASTROPHIC DATA LOSS WITH DIFFERENTIATED DISK DEGRADATION AND FAILURE PREDICTION

In this chapter, I leverage our lab's previous findings to propose a differentiated and proactive disk failure management framework.

7.1. Related Works

Disk failure prediction is not a new topic. A number of methods have been proposed in the literature, such as [96, 91, 72, 61, 46, 45, 93, 66] (Please see Section 7.1 for more discussion). However, they rely on a one-size-fits-all prediction model and future disk state data to "predict" (detect actually) all failures, which significantly comprises the accuracy and applicability of those approaches. In the work [52], Song Huang successfully identified different types of disk failures using SMART data.

A number of existing research efforts seek to characterize the distribution of disk failures and discover indicators of impending failures. Gray et al. [48] observed failure rates ranging from 3.3-6% in two large web properties at Microsoft. Schwartz et al. [78] reported a failure rate of 2-6% in the drive population at an Internet Archive. Schroeder and Gibson [76] found that in the field, annual disk replacement rate typically exceeded 1%, with 2-4% common and up to 13% observed on some systems. They presented the per-component failure percentages for three different types of systems and reported a significant overestimation of mean time to failure (MTTF) by manufacturers. Bairavasundaram et al. [27] revealed the potential risk of latent sector errors during RAID reconstruction, which was not predicted in the early RAID reliability model. Xin et al. [92] analyzed the effect of infant mortality on long-term disk failure rates and used hidden Markov models to describe the effect. Pinheiro et al. [68] studied failures of consumer-grade disk drives used in Google's services. They found that most SMART attributes correlated with disk failures. Ma et al. [60] analyzed disk failures in EMC (now Dell) data backup systems and found that the count of reallocated sectors correlated strongly with failures. Their findings comply with our results from one of the three failure categories [52]. However, as the information of failure categories is not available, all failure instances are considered together in the preceding works. Little work thoroughly analyzes the degradation process of disk failures.

Gunawi et al. [50] collected 101 reports of fail-slow hard- ware incidents and analyzed the root causes including hard disk failures. Many techniques have been proposed to analyze disk failures using SMART data. Gaber et al. [46] extracted features from several time windows and used compound features to reduce the "false alarm" rate for failure detection. Murry et al. [66] compared the performance of support vector machine (SVM), unsupervised clustering, and two non-parametric statistical tests (rank-sum and reverse arrangements tests) on 369 hard drives. They found that the rank-sum method achieved the best performance, i.e., 33.2% FDR and 0.5% FAR. Markov Models [45, 96], probability analysis [61], regression trees [57], and Mahalanobis distance [88] have been proposed to predict disk failures. A number of machine learning techniques were applied to predict sector errors in [27]. Botezatu et al. [30] presented data analytics results on Backblaze dataset to plan the replacement of drives. Rinco n et al. [72] identified 52% of disk failures and achieved a higher performance than the prior work. Xu et al. [93] developed a ranking-based machine learning model to characterize the faulty disks and rank the disks based on their error-proneness. Xiao et al. [91] proposed an online Random Forests model which can be fed in with sequential of SMART data on-the-fly to detect disk failures.

7.2. Critical SMART Attributes and Disk Failure Types

7.2.1. Disk SMART Dataset

Figure 7.1 shows the top eight attributes having the highest correlation with other attributes. Figure 1(a) shows that, among 28.8% of disks, the correlation between RSC and RUE which are higher than 0.75; and in 25.2% of disks, the correlation between RSC and POH are higher than 0.75. Also, in Figure 1(b) and 1(c), RUE are highly correlated with RSC in 41.7% of disks, and POH are highly correlated with RSC in 41.2% of disks. These statistical measurements indicate that RSC is most relevant to other attributes, and RUE and POH follow behind. Hence, the changes of RUE, POH may enlarge the variation of RSC with high possibility, and RSC, RUE, and POH are more sensitive than other attributes as well. In our recent study [52], I found POH was a key attribute for logical disk failures.

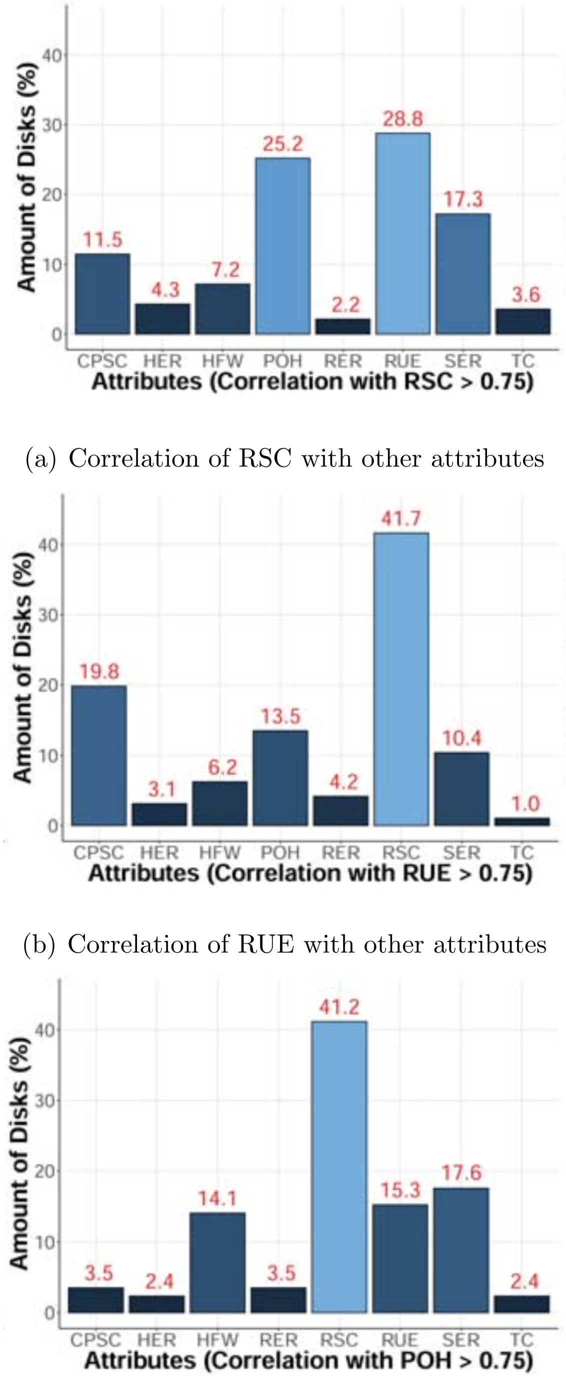
Some SMART attributes, such as “Read Error Rate (RER)”, show high fluctuation when disks are actually in degradation and even close to failures. Their values are not cumulative, that is their readings are either instantaneous or refreshed after a period of time. Thus, they do not maintain the history of the attributes. The high fluctuation can distract our analysis of disk degradation and make the derived degradation signature inaccurate. Therefore, they are not selected as critical attributes.

Data center environmental parameters such as temperature and humidity have been associated with disk failures in the past. This study does not necessarily consider the effect of these factors due to lack of accurate and reliable data points. I note that these factors are causes and may result in observable symptoms considered in this study such as disk read/write/seek errors which may lead to failures.

7.2.2. Understanding Logical Disk Failures

In [52], I explored machine learning and statistical analysis methods to discover types of disk failures from SMART data. Three categories of disk failures, i.e., logical failures (59.6%), bad sector failures (7.6%), and read/write head failures (32.8%), were identified. An important finding was that logical failures account for a large portion of disk failures. These “failed” drives had SMART readings similar to those of working drives, i.e., they did not display physical problems. Those “failures” were more likely caused by errors of the OS or file system that made those drives inaccessible. I referred them to logical failures and used regression trees to model disk health degradation status.

Two important questions that are left unanswered are 1) Why do those logical failures happen? and 2) How can I solve their problems so that those drives can still be used without replacement? To answer these two questions, I conduct a deeper analysis of the logical disk failures.



(a) Correlation of RSC with other attributes

(b) Correlation of RUE with other attributes

(c) Correlation of POH with other attributes

Figure 7.1. Correlation between SMART Attributes among Failed Disks.

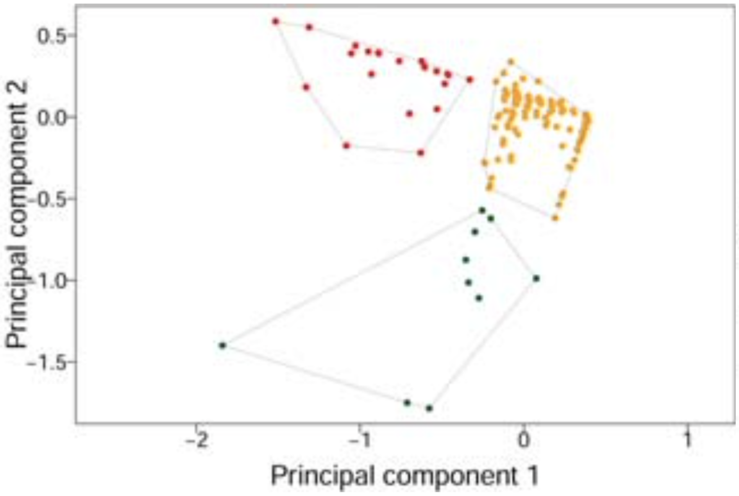
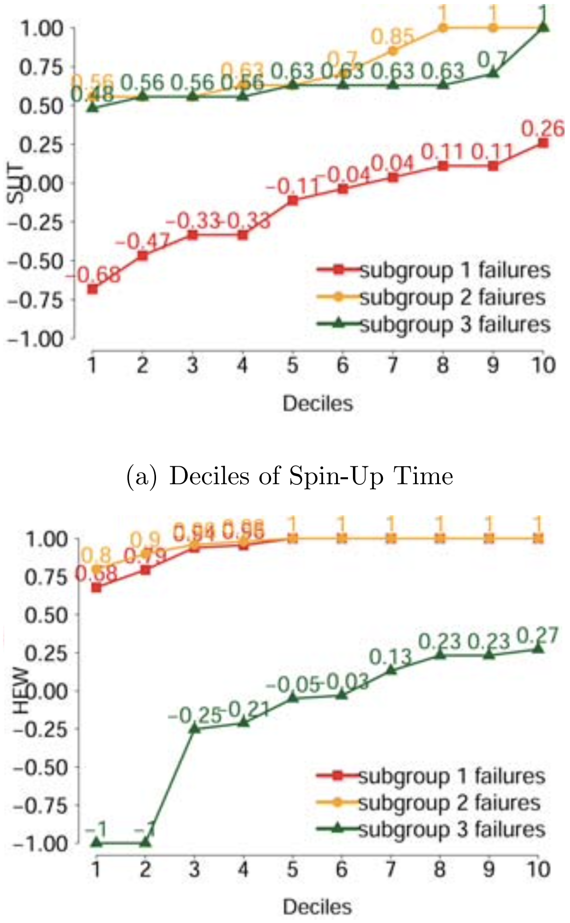


Figure 7.2. Subgroups of Failed Disks with Logical Failures

I leverage data clustering algorithms and statistical analysis methods to investigate the 258 drives having logical failures. Figure 7.2 shows the results from using K-Means++ clustering. Three subgroups are found. The decile distributions of the critical SMART attributes are depicted in Figure 3(a) and 3(b). In Figure 7.2, 8.6% of logical failures are associated with long Spin-Up Time (SUT) (3(a)), which measures an average time used by the spindle to rotate disk platters to the fully operational speed; 4.3% of logical failures are related to abnormal High Fly Write (HFW)(3(b)), which keeps track whether the Read/Write heads are above the normal range during write operations. The rest of logical failures are difficult to distinguish using SMART attributes. Together with the failure types that I have found in [52], five types of disk failures are discovered.

- **RUE failures**, a.k.a. Read/Write head failures, characterized by Reported Uncorrectable Errors (RUE). Uncorrectable errors occur during data communication between disk components and between disks and I/O controllers. Once the number of uncorrectable errors reaches a limit, a disk fails.
- **RSC failures**, a.k.a. bad sector failures, characterized by Reallocated Sector Count (RSC). When an excessive number of bad sectors exist, a disk is not writable.
- **SUT logical failures**, characterized by Spin-Up Time (SUT). It takes a longer than normal period of time to spin up disk platters to the full speed.

- **HFW logical failures**, characterized by High Fly Write (HFW). The Read/Write heads are higher than normal away from the surface of platters during write operations.
- **Other logical failures**. There is no obvious physical damage on a disk, but have a small number of write errors and a small to medium number of read errors. They cannot be characterized by one or a combination of SMART attributes.



(a) Deciles of Spin-Up Time

(b) Deciles of High Fly Writes

FIGURE 7.3. Subcategories of Logical Disk Failures.

7.3. Discovering and Differentiating Disk Degradation Signatures

SMART records the statistics of disk operations and errors. It does not tell if a drive fails or has performance degradation. Without discovering the entire disk degradation process, any modeling cannot accurately characterize this process. To address this problem, I must find the start of disk degradation, which itself is a challenge.

7.3.1. Identify the Start of Disk Degradation

The analysis of “failed”drives (Section 7.2.2) can only tell us how a drive behaves at and close to the time when it is replaced. It cannot show us how the drive change from being healthy to becoming degraded. To find that lost part of the picture, I include those “good” (or called working) drives into the analysis. A drive is working does not mean that it has no erroneous or abnormal behavior. It functions, that is reading data from or writing data to the sectors specified by I/O requests or having spare sectors to replace bad sectors.

I analyze the working drives with an aim to identify those drives that have erroneous behavior from the healthy ones, and thus to discover the start of disk degradation. To this end, I extract the last SMART record of each working drive, and use data clustering, such as K-Means++, algorithms to group the drives into different categories. In my experiment, I find six groups from data clustering. By analyzing the deciles of each attribute for the six groups, I find that five groups of the working drives match the five groups of the failed drives (described in Section 7.2.2), which indicates that the working drives in these five groups have the signs of health degradation. The remaining one is the largest group and does not match with any of the five failure groups with regard to the decile distributions of the critical SMART attributes. It corresponds to the group of healthy drives.

Additionally, I use all SMART records of the working drives in data clustering to verify the preceding finding. The results are the same. No matter whether I use the last SMART record of each working drive or all of their SMART records, data clusters produce the same number of groups with similar decile distributions.

(1) *Read/Write Head Failures*: To determine if a drive degrades with Read/Write head failures, I extract the SMART records of working disks and failed disks which are related to the Read/Write head failures. Then I calculate the RUE deciles of these records and quantify the difference between these records. Figure 7.4 shows the RUE deciles of disk records in three groups: healthy disk records, working disk records

with RUE degrading trend, and failed disks with RUE failures. In Figure 7.4, the working disks with RUE degrading trend, and the failed disks have lower RUE values, while healthy disks have much higher RUE values. I use the upper bound, i.e., 0.37, as the RUE degradation threshold, which means a disk is considered as in a degradation process after its RUE value becomes less than 0.37. For those disks whose RUE sharply drops to 0.37 in a short period of time, immediate replacement is deemed necessary. Drives' health degrades gradually and the thresholds are defined to confirm that drives start to experience health degradation.

(2) *Bad Sector Failures*: Similarly, to identify the start of disk degradation for bad sector failures, I extract SMART records from failed disks having bad sector failures, SMART records from working disks with RSC degradation, and healthy disks. In Figure 7.5, the healthy disks have higher RSC values and the RSC values of 80% of healthy disks are close to 1. In contrast, disks with bad sector failures and disks with RSC degradation have much lower RSC values. I set a hyperplane to separate the disk SMART records into different groups and define the degradation threshold. From Figure 7.5, I choose 0.03 as the threshold for RSC degradation. When the RSC value of a disk drops to 0.03 or less, then the disk is considered as in the degradation process. I can forecast when the disk is going to fail (Section 7.4).

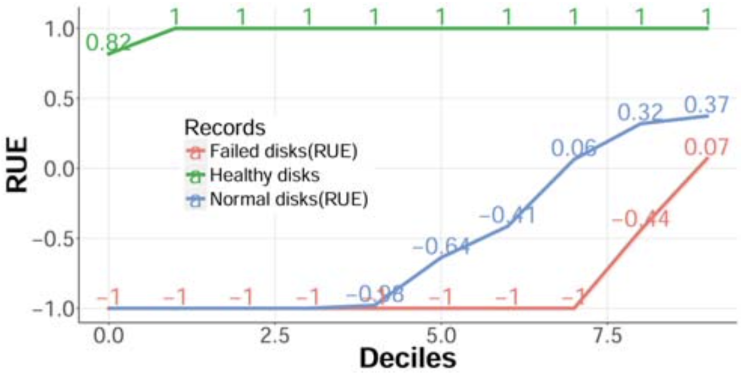


Figure 7.4. Deciles of RUE for Finding the Start of Degradation.

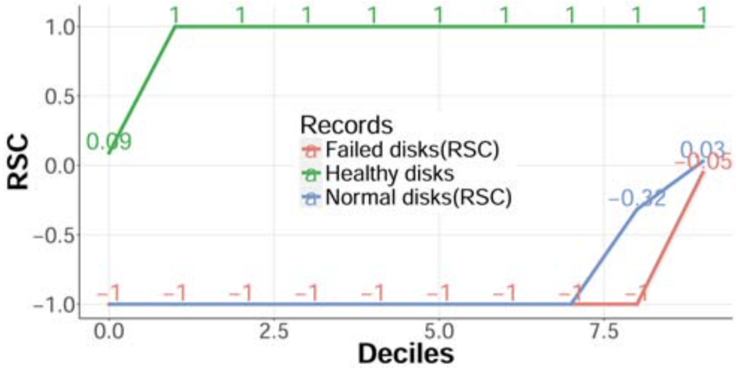


FIGURE 7.5. Deciles of RSC for Disks for Finding the Start of Degradation.

7.3.2. Differentiated Disk Degradation Signatures

After the start of disk degradation is identified, I can discover the entire degradation process from the SMART data of a failed drive. I use a degradation signature to model the degradation process, which is useful for disk degradation monitoring and failure prediction. For each type of disk failures, I derive a degradation signature to characterize the unique trend of disk degradation in that type. In degradation modeling and failure prediction (Section 7.4), I use SMART records from 70% of the failed drives as a training dataset and SMART records from the rest failed drives as a testing dataset for verification.

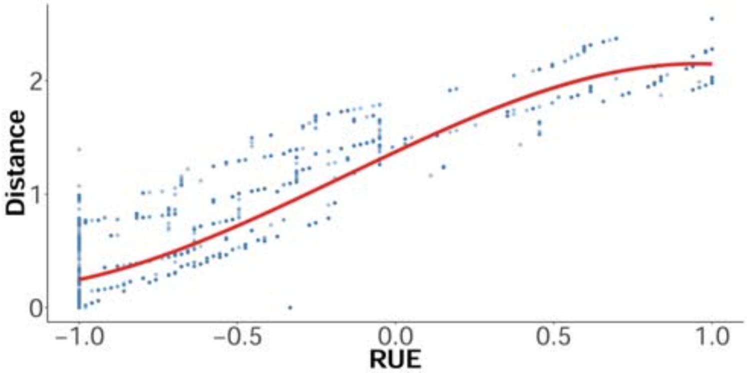


FIGURE 7.6. Relation between Degradation Distance and RUE for Read/Write Head failures (Red Line is the Regression Model.)

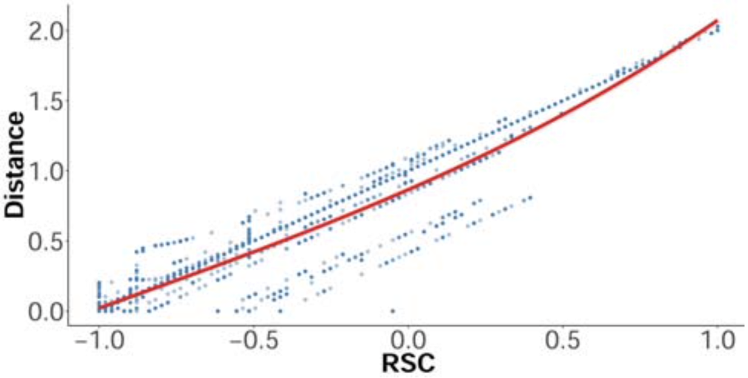


FIGURE 7.7. Relation between Degradation Distance and RSC for Bad Sector Failures (Red Line is the Regression Model.)

(1) *Read/Write Head Failures*: In order to predict future Read/Write head failures, I need to quantify the relation between the degradation status and the critical SMART attributes. For each failed disk, I extract the last SMART record and use it as the failure record of the disk. Then I calculate the dissimilarity modeled by distance, such as Euclidean distance, between each SMART record prior to the failure and the failure record. The distance is converted to a degradation value by using a linear transformation. I build a regression model to quantify the degradation distance, RUE values and other SMART attributes. Figure 7.6 shows the relation between disk degradation and RUE values. Failed disks with RUE failures have a similar relation, that is a low degradation value is mostly caused by a low RUE value. The degradation values can be modeled by the following degradation signature. In Equation (1), disk degradation is determined by changes of RUE and RSC of a disk. For failed drives with Read/Write head failures, the RSC values of 84.6% disks change with RUE, which indicates that RUE impacts RSC with a high prob- ability and both of them should be included in the degradation signature, i.e., Equation (1). To evaluate this regression model, I calculate the R2 and the root mean square error (RMSE) of this model (0.91 and 0.098 respectively), which prove the high performance of this model.

(2) Bad Sector Failures: Following a similar approach, I discover the relation between the degradation distance and the critical SMART attributes, as shown in Figure 7.7. A regression method is used to quantify this dependency. The degradation signature is as follows. The degradation distance is highly correlated with RSC, much more than any other SMART attribute. The red line in Figure 7.7 shows the degradation signature, i.e., Equation (2), that best fits the data points. R2 and RMSE is 0.98 and 0.116 respectively.

$$Degradation = 1.26 + 1.03 * RUE - 0.15 * RUE^2 - 0.28 * RUE^3 + 0.29 * RSC \tag{1}$$

$$Degradation = 0.87 + 0.96 * RSC + 0.18 * RSC^2 + 0.06 * RSC^3 \tag{2}$$

7.4. Disk failures Prediction and Logical Failure Remediation

With a period of monitoring, the collected SMART data can be used to predict when a disk will fail, which is useful for proactive failure management to avoid disk rebuild and data loss. The degradation status, calculated by the degradation signatures, in a sliding monitoring window provides a time series that keeps tracking the change of the disk's health. By using an ensemble of failure predictors, I can forecast the type of disk failure and explore the shape of the corresponding degradation signature to predict when the drive will fail.

7.4.1. Prediction of Two Types of Disk Failures

Figure 7.8 shows the degradation process of a disk, which use the first 50% degradation values (as the sliding monitoring window with history data) for degradation monitoring and predict the rest degradation values using a linear regression model. The blue curve is the actual observed degradation distance. The green curve is the degradation value calculated by the degradation model, with SMART data as input. The red curve shows the prediction of degradation status. Figure 7.8 shows the predicted failure time is a little before the actual failure time.

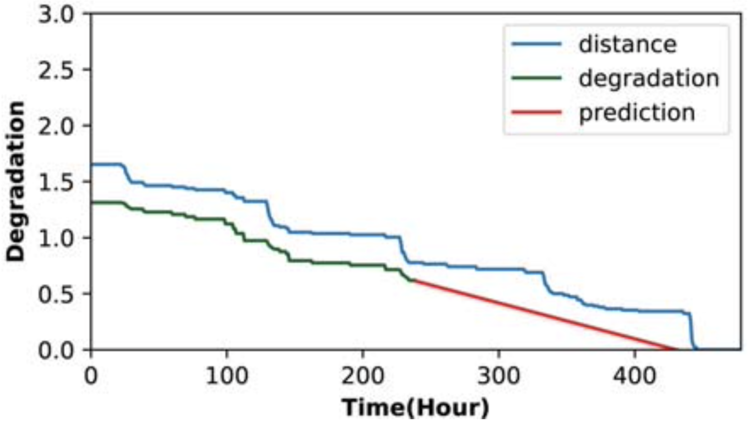


Figure 7.8. Degradation Prediction for Disks with Read/Write Head Failures.

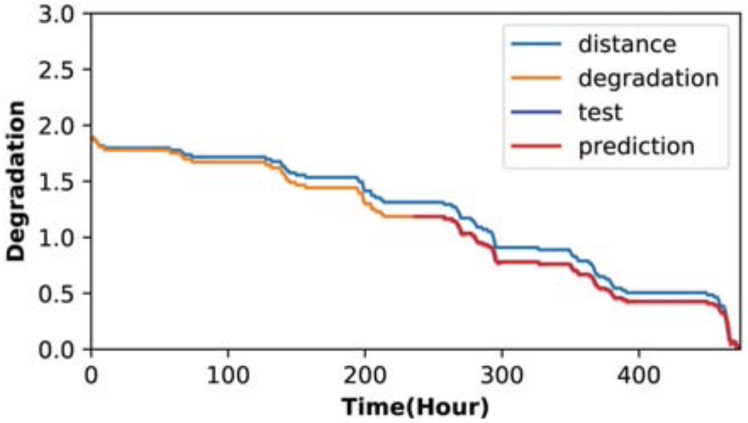


Figure 7.9. Degradation Prediction for Disks with Bad Sector Failures.

As I discussed in the previous section, the RSC value degrades periodically and manifests as "step" pattern. It is not easy to predict the failures using linear regression model. Hence, I apply Auto-Regressive Integrated Moving Average (ARIMA) model to forecast the degradation values as time series data. ARIMA model not only consider the trend of the time series data, but also the seasonal, cyclical, and irregular components, which means it accounts for the relationship through time. I consider the contribution of the RSC values in the previous periods to the current RSC value, and converting a non-stationary to a stationary process by differentiating the time series, to quantify the current RSC values.

Figure 7.9 shows the prediction results of using the first 50% of time series data as monitoring data. It displays the actual distance to the failure point of the disks (actual degradation status), the mapped degradation status (calculated from the degradation model), the degradation test data and the prediction results on the test data. In Figure 7.9, the blue curve which represents the observed degradation data is mostly covered by the red curve which represents the prediction results. This indicates that the prediction results are accurate as the difference between the actual failure time and the predicted failure time is less than 3 hours.

I deploy the degradation signatures and prediction models to a storage system on campus, to calculate the degradation status and forecast the upcoming failure time of disks. I find that the difference between actual failure time and the predicted failure time of RUE failure is less than 20 hours, and the difference of RSC failure is less than 8 hours. A key factor of failure prediction is the accuracy of the degradation signatures based on the SMART attributes, because the ARIMA model shows its outstanding performance on prediction.

Compared with our previous work [52] that used regression tree to estimate disk degradation status, our proposed performance signature-centric approach achieves a higher prediction accuracy and reduces the root mean square error rate for Read/Write Head Failures and Bad Sector Failures to 0.098 (i.e., by 14.03%) and 0.116 (by 10.07%), respectively.

Compared with the long rebuild time and simultaneous failures in a disk group during rebuild which causes data loss, our proactive method is safer and more effective. The data transfer rate of modern hard disk drives with an eSATA interface reaches 150 Mbps [74]. The data backup overhead of a full 4TB disk can be finished around eight hours. Therefore, right after the start of degradation of a disk drive is detected, our methods keep monitoring the degradation dynamics and forecast the failure time with a high accuracy, which enables the system administrators and management tools to have a Lead Time to Failure (LTTF) long enough to migrate the data to a healthy, spare disk.

7.4.2. Remediation of Logical Failures

Logical failures are caused by software errors. The disk drives display no obvious physical damage. If the root-cause errors can be corrected, these disk drives can continue functioning without replacement, which reduces both the repair cost and disk rebuild overhead. In this section, I study two types of logical failures and discuss possible remediation.

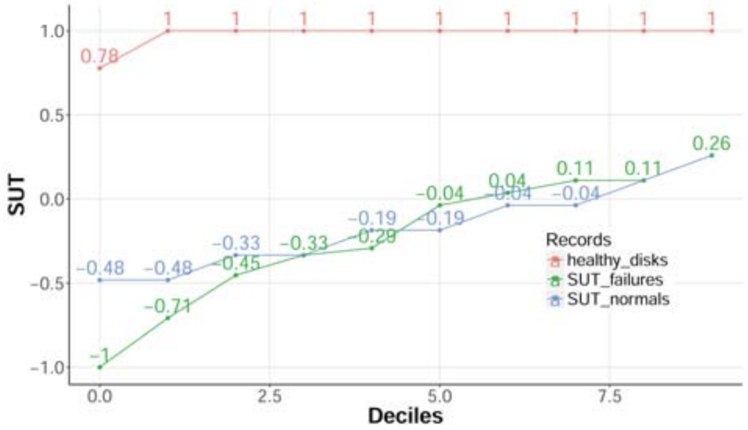


FIGURE 7.10. Attribute Distribution of the SUT Logical Failures.

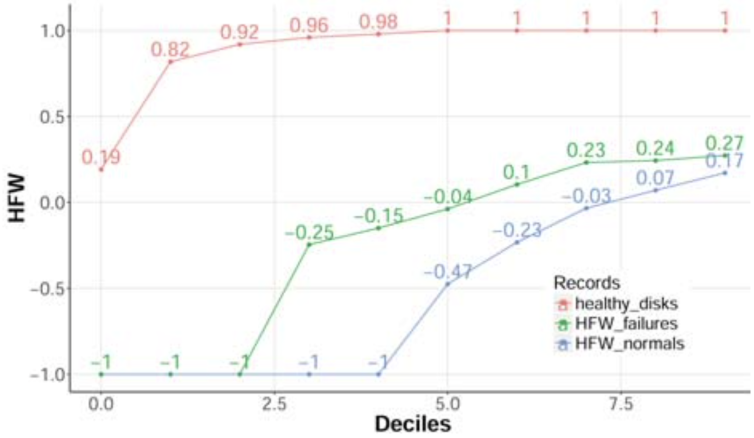


FIGURE 7.11. Attribute Distribution of the HFW Logical Failures.

(1) *SUT Logical Failures*: Spin-up-time (SUT) is the amount of time for the platters to reach an operational rotating speed. Longer SUT is usually considered as caused by hardware problems, either the spindle itself or the bearing has wear and tear. However, the SUT-related disk failures that I discover in Section 7.2.2 are a subset

of logical failures. After consulting with storage experts, I find out that the power saving techniques can also affect SUT. When the I/O workload decreases, that is a less number of disk reads/writes, modern hard disk drives switch to the power-saving mode(s), in which the rotation speed of disks drops. When a burst of disk read/write requests come, the spindle of the disks takes extra time to reach the operational speed. The disk drives have no physical problem. The power management software causes this extra spin-up time. To remedy this SUT-related logical disk failure, I can balance I/O workload over time so that there is no obvious disk idle time. This remediation needs the assistance of I/O nodes which distribute I/O requests to disks. Another way is to change the power saving strategy to keep disks always rotating at the operational speed. However, it leads to more power and energy consumption. Thus, the trade-off among disk reliability, control complexity, and energy consumption needs to be considered to address this type of disk failure. Predicting SUT failures is not an easy task because the SUT value has a long degradation process. The time series of SUT is usually flat. Figure 7.10 shows the decile of healthy disks and working disks with an SUT degradation trend, and that of failed disks with SUT-related failures. The healthy disks can be distinguished from those disks with SUT degradation trend and disks with SUT-related failures using a hyperplane of SUT. From Figure 7.10, I find that 0.26 is the degradation threshold for SUT values. When the SUT reading from SMART is less than this threshold, I can apply the preceding remediation action to prevent the logical failures.

(2) *HFW Logical Failures*: When data is written to disks, the High Fly Monitor provides protection for write operations by recording the distance between the read/write head and platters. If the flying height is beyond the normal range, a high HFW value is recorded. As for logical disk failures, I study the specifications of disk drives and online documents and discussion groups of disk manufacturers, and find out that a thick air cushion between the read/write head and disk platter is a cause of abnormal HFW values. Normally, the air cushion prevents the read/write head

touches a disk platter. However, if the read/write head stays at a track for a long time, the rotation of the disk causes more air to accumulate between the track and the read/write head, which produces a High-Fly- Write error. To remedy the HFW-related logical disk failures, I can distribute the data access across different tracks. Thus, the read/write heads would not stay at a track for a long time to trap too much air. To realize this, the file system or operating system may distribute hot and cold data evenly on disk drives. Another way to tackle this type of failure is to move the read/write head to other tracks when the HFW reading is detected as anomalous. This requires the operating system to monitor the SMART readings. Similar to SUT, the time series of HFW has a flat degradation curve and a long degradation process. In some cases, the value of HFW may go up a little and then drop again. This reflects the variation of the thickness of air cushion between the read/write head and disk platters. Thus, it is difficult to predict the HFW-related disk failures by using the time series of HFW. To determine whether disk drive may have HFW-related failures, I quantify the degradation threshold from HFW readings. Figure 7.11 shows the HFW deciles of healthy disks, working disks with HFW degradation, and the failed disks with HFW-related failures. I find that 0.27 can be used as a threshold to identify the start of the degradation process for HFW-related failures. When the monitored HFW reading is below the threshold, our proposed remediation action(s) can be applied to prevent HFW-related logical failures.

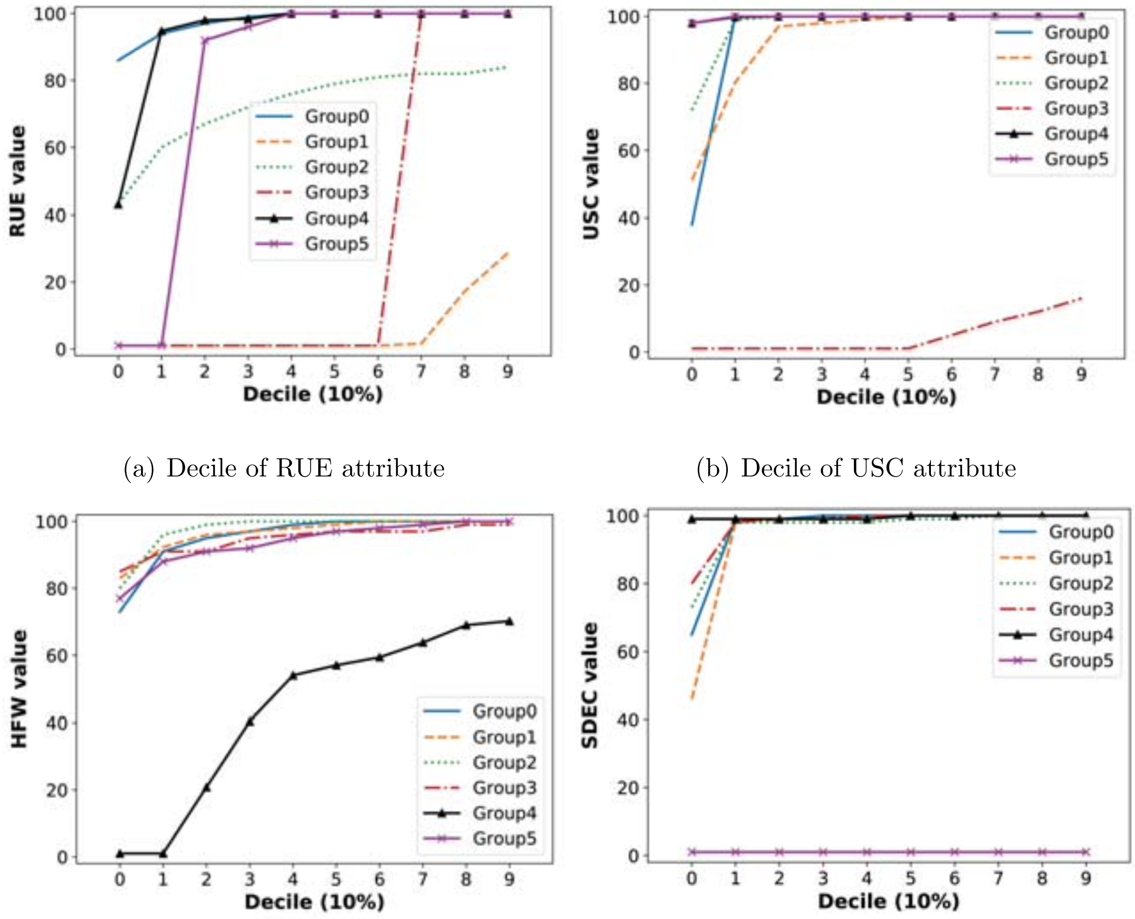
7.5. Method Generalization and Verification: Experimental Results Using Backblaze Dataset

According to the observations, there are some slight differences between the collected SMART attributes of disks which are from different manufacturers or different models. Also the distributions of the same attributes may not be the same. Some distribution bias or shift exists in the disks from different branches. Therefore, one of our targets is to generalize our methods to other storage systems even running different workloads. I apply our methods to the Backblaze dataset which is mentioned in Section 7.2.1, I found that the failures can be categorized into 6 different groups: RUE-related failures (can be divided into Group 1 and Group 2), Uncorrectable Sector Count(USC)-related failures (Group 3), HFW-related failures (Group 4), SATA Downshift Error Count (SDEC)- related failures (Group 5), and logical failures (Group 0). I scale the value of attributes to the range of [0, 100]. Figure 12(a), 12(b), 12(c), 12(d) show the decile of RUE, USC, HFW, and SDEC attributes. The RUE-related failures are divided into 2 groups because some of them failed when the RUE are 0, and some of them failed when RUE values drop down to 50%. This results from the one group of failures are impacted by other attributes as well. RUE-related failures have relatively long degradation process and HFW-related failures have even longer degradation process. Use the same method as before, I can calculate the thresholds to determine the starting point of degradation. On the other hand, USC-related failures and SDEC-related failure degrade very fast, and setting the thresholds is not enough. The system administrators need to monitor the degradation process once the USC and SDEC value begin to change.

For RUE-related failures and HFW-related failures, which do not degrade abruptly, I characterize the degradation processes by building an Attribute-Health Status Regression model to eliminate the dependency on workload. It maps the SMART attributes to the degradation status. The orange lines in Figure 7.13 and 7.14 are predicted degradation status from the regression model based on the attribute values. I can see that the predicted status is quite closed to the actual degradation status which are calculated by Euclidean distance method, especially when the disks are degrading and closed to the failure point. I evaluate the overall performance of these regression models by calculating the Root Mean Square Error, which are 0.1839 and 0.1624 respectively.

Finally, I apply the Time Series prediction model to forecast the upcoming failures. Figure 7.13 and 7.14 display the prediction results on RUE-related failures and HFW-related failures. I use the sliding window model by feeding in some history records to improve the prediction accuracy. The green lines in the figures are the prediction results which are closed to the blue lines (actual degradation status), and the orange lines (the degradation status generated from the regression models), most of them are even overlapped. It means the predicted results are accuracy. Further more, the results from both production datasets prove that, the our prediction method can forecast the future failure accurately.

Our proposed method is lightweight. It takes less than a few seconds to cluster the SMART records of all failed drives, calculate distributions, and build degradation signatures on an Intel Xeon E5-2683 machine. Disk failure prediction for a drive takes about 0.6 second for a prediction window.



(a) Decile of RUE attribute

(b) Decile of USC attribute

(c) Dfsioe-Read Read/Write Request

(d) Decile of SDEC attribute

Figure 7.12. Deciles of Critical Attributes

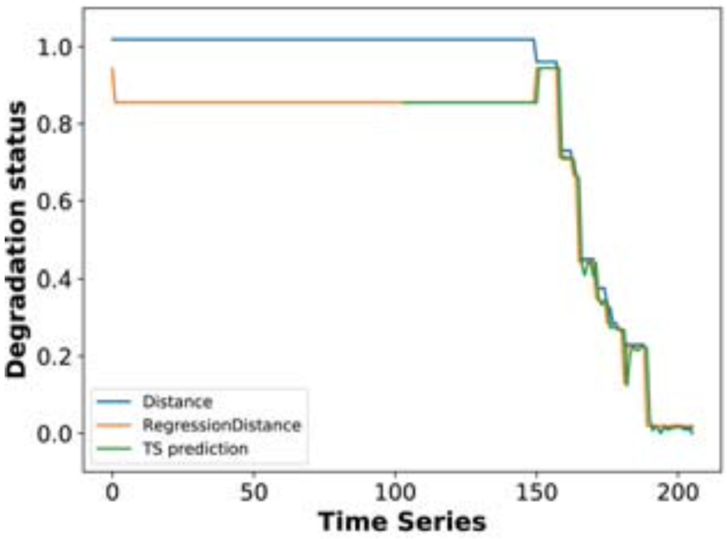


FIGURE 7.13. Regression Model and Prediction of RUE-related Failure.

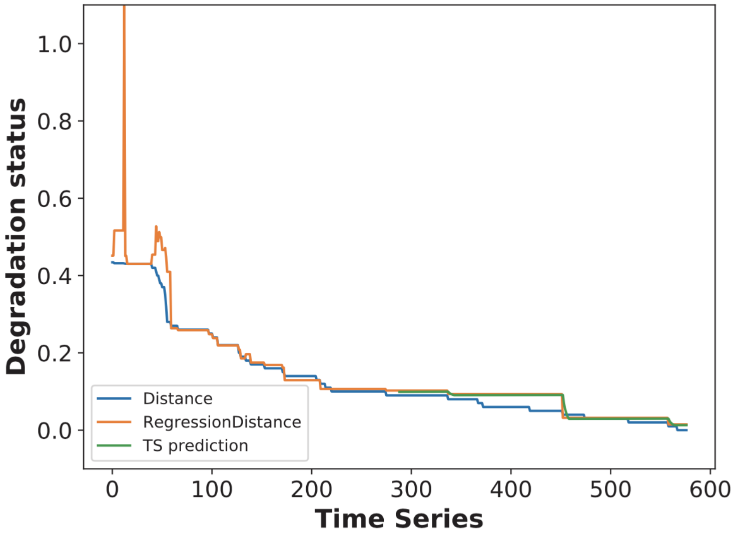


FIGURE 7.14. Regression Model and Prediction of HFW-related Failure.

CHAPTER 8

ENABLE ON-DRIVE RELIABILITY MANAGEMENT VIA OPEN ETHERNET DRIVE

8.1. Introduction

It has been nearly a decade since the increase of CPU frequency fall behind the Moore's law. Instead of slowing the computational power improvement, innovations like domain-specific hardware (e.g., TPU, Crypto-currency Miner, etc.), enhanced security, open instruction sets, and agile chip development lead the way to a new golden age for computer architecture. As a result, the aggregated compute power for AI and big data computing still increases at a rate following the Moore's law due to these heterogeneous computing advancement. Novel computing paradigms have been proposed to offload computation from traditional multi-core CPUs to many-core processor such as GPU or TPU, and even to computer peripherals such as memory, disk drives, and NIC (i.e., in-memory processing, smart drives, and edge-computing enabled network controllers).

Storage systems are indispensable for big data processing today. The ever-growing size of computation and data analytic results demands larger storage capacity, which challenges data processing and storage scalability. Moreover, the increasing complexity of storage hierarchy and "passive" storage devices make today's storage systems inefficient, which necessitates the adoption of new storage technologies. In short, OED is a hard drive that can do computing on drive itself, rather than moving data back and forth to main CPU.

In this section, I demonstrate the advantage of on-drive reliability management using OED. I set up a test environment that consists of a host Linux server and two OED drives. The host can communicate with each OED via serial port and SSH channel over 1 Gb Ethernet connection. Each OED runs embedded Linux Debian OS that is self-contained and works as a regular Linux server. I have conducted comprehensive experimental evaluation to measure the performance and power consumption of OED and compare it with traditional server. In addition to I/O operations, I test and evaluate on-drive data processing, including data compression, aggregation and erasure encoding, which provide natural support for data-intensive applications. The evaluation provides first-hand results of this new hardware and demonstrates its superior benefit for energy-efficient computing.

To build a storage cluster with Ethernet-enabled micro storage servers, I need a reliable mechanism to orchestrate the data distribution among a large number of Open Ethernet Drives (OED). My initial plan was to adopt the manager-worker model that uses a centralized manager to control and manage all the OED drives. However, due to the norm of disk failures and various datacenter service level agreements (SLA), assigning an OED as the manager could lead to a single point of failure in the cluster. Thus I design a decentralized scheme exploring distributed hash tables (DHT) or peer-to-peer (P2P) network. I have evaluated several software for our needs such as OpenDHT by UC-Berkeley, Bamboo DHT by UC-Berkeley, Open Chord by MIT, FreePastry by Rice, Chimera/Tapestry by UCSB, and Kademlia DHT by NYU. Due to the high run-time resource demand and high latency by JRE, I excluded software that implemented in JAVA. I tested all candidates using C++ implementation, only Kademlia works for the OED environment. Others require many dependencies and libraries that are not compatible with the ARM processors used in OED drives.

I setup an active storage cluster consisting of a host server and 60 OED drives. Each unit in the cluster shares the name space over a network file system (NFS). I use parallel Erasure coding (ParEC) to distribute erasure coded blocks. For traditional storage systems using hard disk drives, host servers handle erasure encoding, decoding and orchestrate data distribution on individual drives. In contrast, hosts in active storage systems only assign source file blocks to corresponding OED drives. The additional encoding, decoding, and erasure code block distribution are handled by the OED cluster. Compared with the traditional design, active storage systems relieve the hosts from data-intensive computation, which also reduces the power consumption and carbon footprint. Our experimental results show that Open Ethernet Drive can significantly lower the energy consumption while maintaining the data processing throughput simultaneously by ensuring data availability and storage scalability. Results and findings from this work will facilitate scheduling of on-drive compute resource for building active and scalable storage systems.

8.2. Incorporate Near-data computation for Reliable Data Storage

As the size of data centers continues to grow, many issues, such as scalability, performance, reliability and power consumption, arise. In this paper,I introduce ACTOR, an active storage paradigm to address two of them: storage resilience and energy efficiency. When exabytes of data to be stored on hundreds of thousands of disk drives, storage reliability becomes a major challenge. At such a scale, data corruption and disk failures become the norm. For large-capacity (e.g., 8-12 TB) hard disk drives, it could take several days or even a week to rebuild a failed drive. In a large storage system with hundreds of disk arrays, more drives may fail when a drive is being rebuilt, which results in significant performance degradation and escalates the risk of data loss. Dealing with data corruption and disk failures becomes the standard procedure for data center operation.

Energy consumption is also an important issue for operating data centers. The storage subsystem in HPC and big data systems consumes around 20% of the total power usage. Likewise, it also accounts for up to 50% of the total cost of ownership for data centers. A recent field study[40][55] reveals that many applications exhibit burst I/O accesses in contrast to consistent storage access pattern throughout a day. Much power is wasted when the workload level is low. Energy efficiency is a major concern for large-scale storage.

I explores Ethernet-connected Drives to develop parallel erasure coding[32] and build the active, scalable, resilient, and energy efficient storage paradigm (shown in Figure 8.1). By the time of this study, there are three major models of Ethernet connected disk drives in the market that provide comparable functionality. I compare them with a close examination of their advantages, as listed in Table 8.1. These products possess attractive features such as: 1) embedded ARM based low-power CPU, 2) on-board memory, 3) running Linux OS, 4) Ethernet access and TCP/IP network connection, and 5) high capacity disk and/or flash storage. These features enable us to offload data processing operations closer to the storage media and reduce system power consumption.

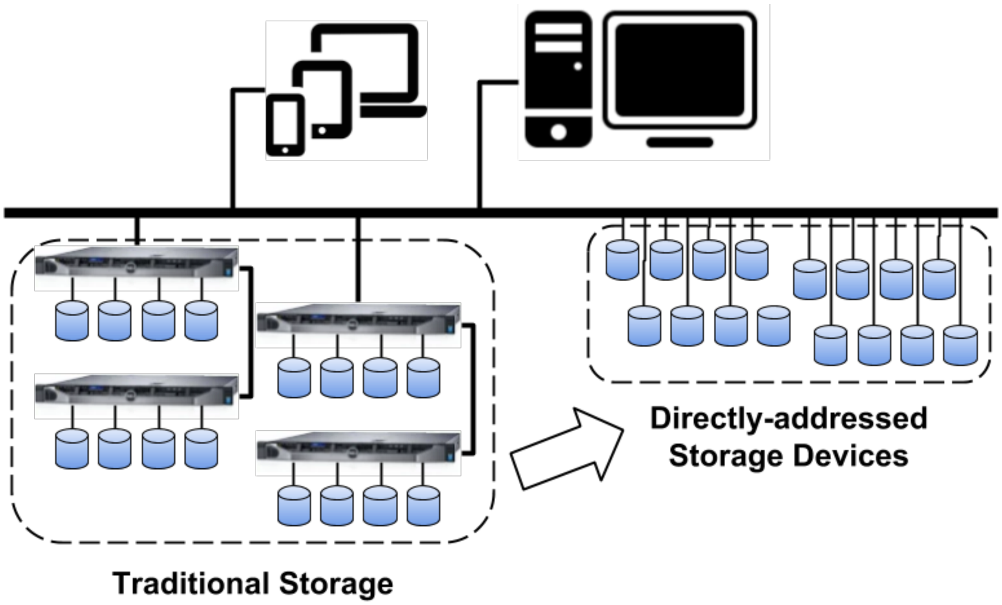


FIGURE 8.1. Shifting from Passive Storage with a Storage Server-Drive Paradigm to a Cluster of Highly-Parallel, Computing-Enabled Autonomic Drives.

TABLE 8.1. Specifications of Ethernet Connected Drives

	Manufacture	Processor	Memory	OS	Storage	Ethernet	Platform
OED gen1	HGST	32-bit ARM 1 GHZ 1-core	2 GB DDR3 1600 MHZ 1-channel	Embedded Debian 7.4	4TB HDD 7200rpm	1 Gbit/s active	Open platform, run applications
OED gen2	HGST	32-bit ARM 2.2 GHZ Dual-core	1 GB DDR3 1600 MHZ 2-channel	Embedded Debian 8.4	8TB HDD 7200 rpm	1 Gbit/s active	Open platform, run applications

8.2.1. Ethernet Connected Drives

As a subsidiary of West Digits (WD), HGST has its own directly-addressed storage disk drives[6]. Open Ethernet Drive (OED) has been designed as a micro-storage server[5]. The first-generation OED has one single-core 32-bit 1 GHz low-power ARM processor, 2 GB DDR3 memory at 1600 MHZ, a Gigabit Ethernet link, and 4 TB of storage space while maintaining a traditional 3.5-inch disk drive form-factor. It runs embedded Debian 8.0 OS with kernel v3.14.3. The first generation OED supports a vendor supplied extension board that can be coupled with OED for development on single drives. The second-generation OED upgrades hardware to accommodate increased need for computation power. It comes with a 32-bit ARM dual-core processor that clocked at 2.2 GHz, Debian 8.1 OS, 1 GB dual-channel DDR3 memory, and 8 TB of storage space. Unlike the 1st-generation OED that can be operated individually, the 2nd-generation OED requires a designated 4U storage enclosure (i.e., JBOD) to operate. Each JBOD enclosure can support up to 60 OEDs and offer 480 TB capacity without redundant configuration. Each hot-swappable OED is connected to the 60 Gbps internal bus, and is accessible via one of the four 10 Gbps external Ethernet ports.

For Ethernet connected drives that provide the KV store interface, using *Get* and *Put* object-style operations simplify application development with ease of moving data between applications and storage. This also requires a broader support from the development community to provide updates of new access protocols, open-source libraries, etc., in order to interact with existing applications. On the other hand, OEDs from HGST provide open access to the embedded OS, so that servers can manage disk drives directly as if they were worker nodes. OEDs enables “data-centric” storage services that can support software-defined storage (SDS). As a result, I choose OEDs as the active storage component for the design of ACTOR.

8.2.2. Incorporating Parallel Erasure Coding in Object Storage

Being able to easily and inexpensively create, manage, administrate, and migrate unstructured data is crucial for big data analytics . It is difficult if not possible for the traditional storage paradigm to meet these requirements, and they suffer from performance degradation when a system scales out[82]. Erasure-coded storage has been widely adopted to provide cost-effective space saving and fault resilience[69]. I integrate erasure-coding techniques with object storage to manage unstructured data as unified redundant data objects in a highly-parallel, scalable OED storage mini-cluster.

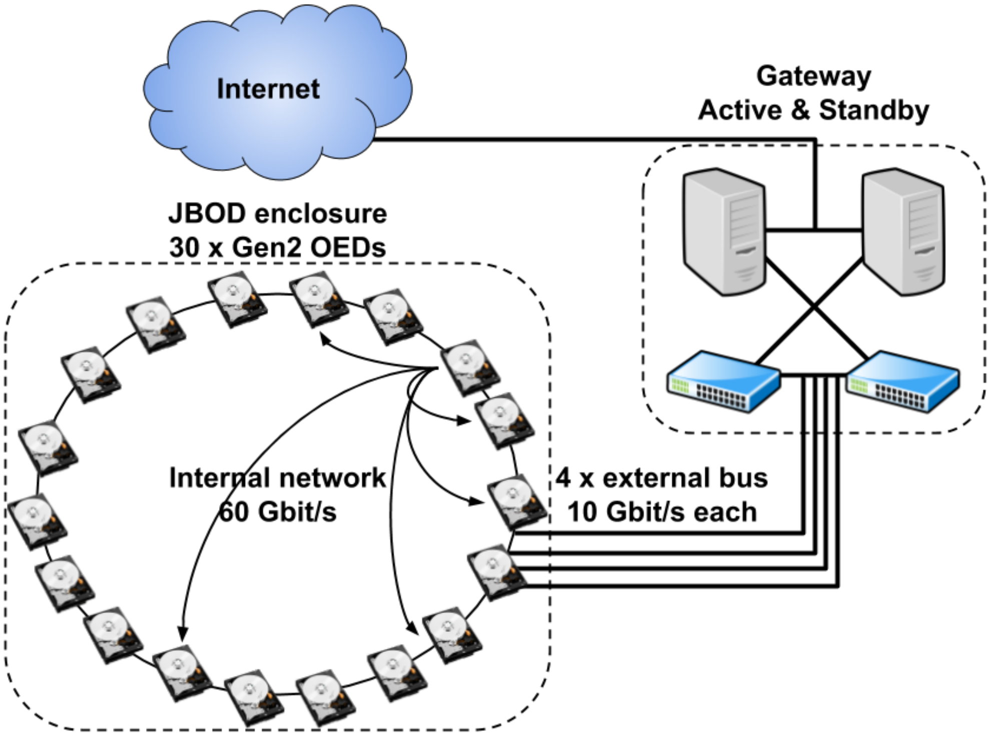


Figure 8.2. Architecture of ACTOR.

8.2.2.1. Parallel Erasure Coding

As an efficient data resilience technology, erasure coding gains an uprising attention . However, existing systems use a single machine or a multi-threading approach to perform encoding and decoding, which is not scalable to process ever-increasing big data. In addition, I notice that *SYSTEM-IDLE* states account for a significant portion of the overall processing time and energy consumption in the erasure coding/decoding process. When encoding and decoding a large file, single machine is not an efficient solution. I propose a Parallel Erasure Coding (ParEC) technique[42] to address such concerns.

In today's processors, multi-core CPU has embedded Single Instruction Multiple Data (SIMD) subsystems. Inspired by[69, 43], I utilize the processor's SIMD extensions for data parallelism. I extend the existing single-process encoding/decoding approach by adding MPI parallel I/O for task parallelism. With the combination of data and task parallelism, ParEC improves the performance and scalability of erasure coding techniques. Moreover, power and energy is saved by minimize the system idle and exploit overlapping between data migration and encoding/decoding process.

8.2.2.2. Data Placement in OED-based Mini-Cluster

At a high level, ACTOR aims to substitute storage servers in the traditional storage system with a mini-cluster of networked OEDs. Gateway servers have direct accesses to each OED drive. However, deserializing unstructured data into objects and vise versa needs to re-engineer the data access protocol. As illustrated in Figure 8.2, I explore distributed hashing method called *ring*, in which each Ethernet drive governs a data region on circular space. For a file with N data segments, ParEC generates K parity code chunks. This $N+K$ encoding scheme can tolerate up to K data segment loss. Upon a large data-set arrives at P2P network, it is deserialized into N segments which mapped to N corresponding OEDs on the ring. These N OEDs then launch ParEC to generate K parity segments, and disseminate them to non-overlapping K OEDs. When client issues a data retrieval request, the N OEDs that initiate the encoding process are responsible for decoding the erasure data objects before returning the data chunk to the client.

OEDs are more than a hard disk drive that uses Ethernet for communication. They provide a higher level of abstraction for data transfer and on-drive data processing. OED can offload traditional logical block address (LBA) management and disperse object data to disk drives. I integrate this new technology into the active storage paradigm to simplify storage system software stack, enable transparent drive-to-drive data migration, and reduce the significant drive read/write traffic from compute nodes.

8.3. Performance Evaluation

The evaluation of ACTOR focuses on two aspects: real-world storage performance and energy consumption for data reliability management tasks. Since ACTOR primarily targets storage reliability management, I measure its performance and energy consumption using Erasure encoding and decoding of ParEC applications. Our evaluation aims to answer the following question: Compared with the existing storage paradigms, **Can ACTOR deliver**

energy efficient reliability management? I measure energy consumption $E(Wh)$ of executing a task as a product of power $P(watts)$ and time $T(hours)$, that is $E = P \times T$. For comparison, I use a server that is equipped with a SATA hard disk drive to represent the passive storage. First, I measure the execution time of each benchmark workload on the server and a single OED to compare T_{server} and T_{OED}. Then, I increase the number (n) of OEDs in ACTOR until its aggregated performance matches with that of the server, where $T_{server} \approx T_{OEDs}$. I measure P_{OEDs} and check if $E_{OEDs} < E_{server}$.

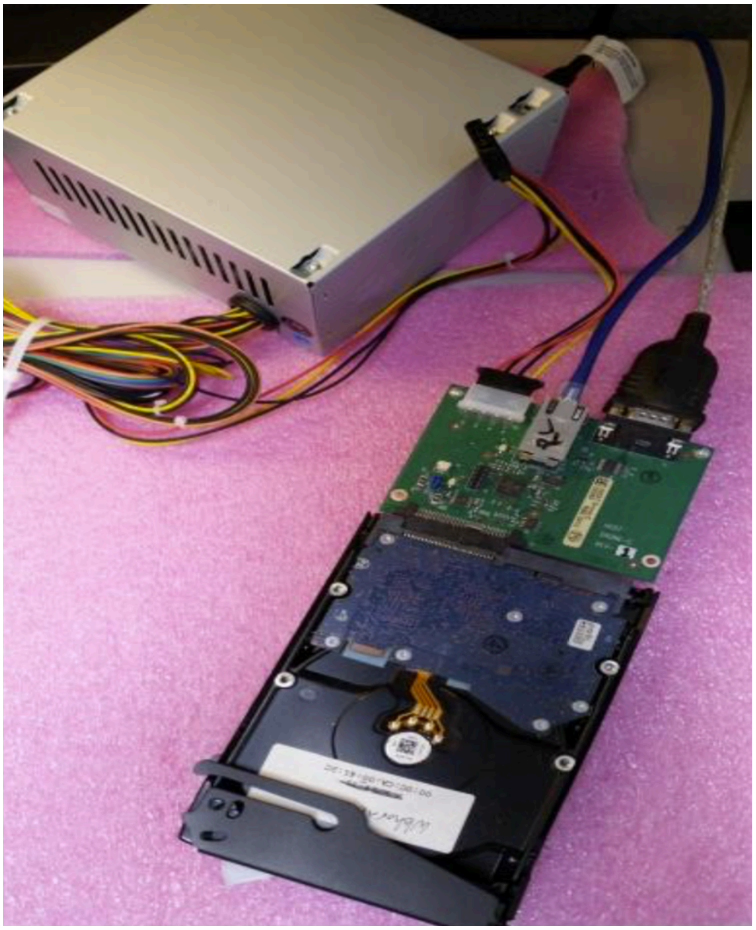


FIGURE 8.3. OED Drive is Connected Directly to Power Supply and Ethernet Switch as a Micro-Storage Server.

8.3.1. Test Platform Configuration

The servers that I use in my experiments are equipped with eight Intel Xeon cores (3 GHz), 8 GB of DDR3 memory, and Seagate BarraCuda ST2000DMs hard disk drive. The

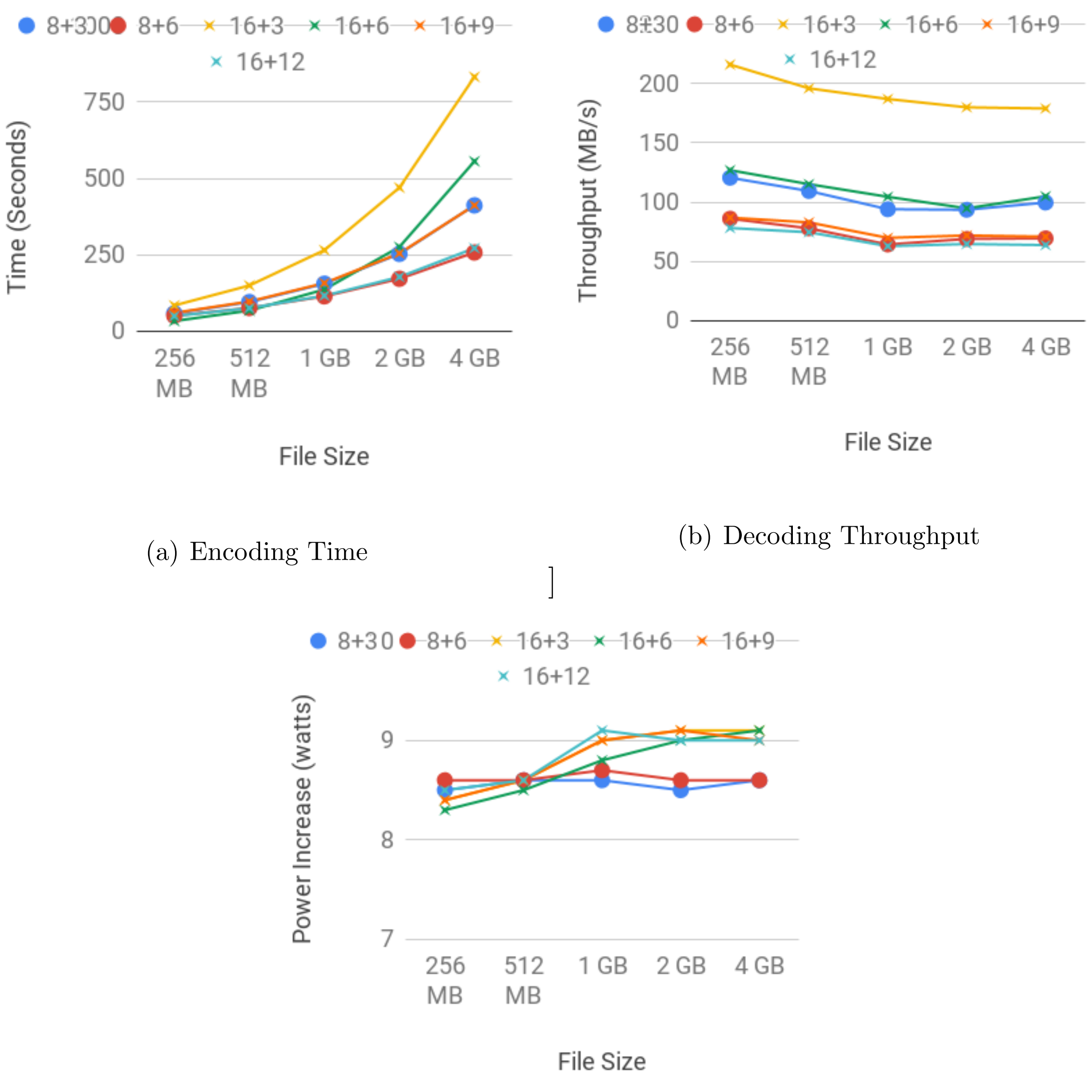
OS is Ubuntu 16.04 LTS. I study both generations of HGST OED drives in the experiments. The first-generation OED uses an extension board as an interface for power supply, Ethernet connection and serial port. The servers and OEDs are connected by Ethernet cable via a Gigabit switch. After activating OED's embedded OS, Debian 7.4 Wheezy, the available memory for user programs is 1.79 GB. Block storage is 4 TB as an SCSI disk (*/dev/sda*). Figure 8.3 shows the setup of the OED drive (first-generation). The appearance of OED is the same as a standard 3.5-inch HDD form factor with extension board which provides necessary external connections through a SATA interface. The second-generation OED drives are installed in a JBOD enclosure at Los Alamos National Laboratory. It consists of 30 OEDs with 10 Gbps Ethernet connection.

Power consumption is measured at room temperature, that is around 76 degree Fahrenheit, and 80% humidity, using *WattsUp Pro* power meters. The accuracy is within 1.5%. Its current and power factor readings are within ±1.5%[8]. Monitored logs are transferred to a desktop computer for power consumption analysis.

8.3.2. Erasure Coding Performance on OED

In this set of experiments, I measure the performance of erasure coding, using Zfec, on OED. Zfec[9] is a popular erasure coding implementation, which expands the size of input data by adding redundant blocks of information (i.e, check bits) to the original data in the encoding mechanism, whereas they are removed in the decoding process. Note that our ParEC data placement algorithms also based on Zfec. Two parameters, K and M, are used, where M represents the number of share files with size as $\frac{1}{K}$ of the input file size.

Figure 4(a) shows the encoding and decoding time of Zfec on OED drive. From $M = 8$ to $M = 16$, higher encoding difficulty takes more time. When $M = 8$ and $K = 6$, the time to encode is less compared to that when $M = 8$ and $K = 3$, because as K increases the number of share files also increases, which decreases the number of check bits added to each share file, thus decreasing the encoding time. Also, when $M = 16$ and increasing the K, encoding time decreases. Similarly, as K increases the time to decode a share file should decrease as well. But interestingly, there is no much time difference with increase in K. The curve steep is almost the same for all K values.



(a) Encoding Time

(b) Decoding Throughput

]

(c) Power Consumption

Figure 8.4. Experimental Results of Erasure Coding with N Data Blocks and K Parity Blocks.

As we have seen earlier, as the number of share files increases the power required to encode the file also increases. Figure 4(c) shows the power consumption of running Zfec erasure encoding on OED. From the figure, it is clear that when $M = 3$ and $K = 16$, it takes more power to encode whereas when $M = 8$ and $K = 6$, it takes less power. The power consumption ranges from 8.4 watt to 8.7 watt in addition to the idle power consumption where $P_{idle} = 16watts$, which is optimal. Similarly, decoding power usage (not shown, but has similar trend) also increases with the number of share files. Likewise, with the increase of file size, power consumption for erasure decoding also increases. After the memory overflow(i.e., file size $\geq$ memory size), the power consumption for decoding remains stable with increased file size. The power consumption ranges from 8.2 watt to 9.2 watt in addition to the idle power consumption where $P_{idle} = 16watts$.

Figure 4(b) presents the measurements of throughput when performing erasure coding on OED drive. I observed that throughput of erasure decoding is almost 10× higher than encoding phase since decoding is faster than encoding. It also shows that for encoding with an increased number of share files, the throughput decreases. So $M = \mathit{8}$ and $K = \mathit{8}$ has a higher throughput compared to $M = \mathit{16}$ and $K = \mathit{3}$. Interestingly, the opposite is true for decoding i.e, the throughput of decoding increases with an increase in share files, e.g., the throughput of $M = \mathit{16}$ and$K = \mathit{3}$ is much greater than that of $M = \mathit{8}$ and $K = \mathit{3}$. Overall, the throughput of erasure encoding and decoding is desirable on OED drive.

8.4. Discussion and Implication

In this section, I present ACTOR, an active Cloud storage paradigm that utilizes the on-drive data processing capability of Ethernet-connected drives as the intelligent Big Data reliability management solution. ACTOR leverages a large number of low power Ethernet-connected drives, which can reduce power consumption and electric bills for big data applications. Moreover, processing data closer to storage media allows us to offload computation and reliability management workload to Ethernet-connected drives, which enhances parallelism and reduces data movement. We can take advantage of this hardware parallelism to handle a massive amount of data sent to the storage system from data-intensive applications. We can further save network bandwidth by performing data compression and data aggregation before data transfer, so much less data will be transferred to compute nodes. I extensively evaluate ACTOR and Ethernet-connected drives with various tools and benchmark suites to measure the performance of data processing and data compression, and storage reliability.

CHAPTER 9

CONCLUSIONS AND FUTURE WORKS

9.1. Conclusions

In this dissertation, I first investigated SSD-specific SMART data collected from an active production datacenter. I find that SSDs have many unique attributes compared with HDDs. By analyzing these SSD-specific attributes, I find that they are very useful for characterizing and modeling the health status SSDs at the device level. The analytic results show that the volume of I/O operations and P/E cycles has a significant impact on the wear level of SSDs. Write and erase related attributes display strong correlations. In a well maintained datacenter, environmental attributes do not have directly influence SSD reliability. I also observe health state transitions which correspond to SSD reliability degradation.

Next, I study the QLC SSD performance, as well as its economic effects on the landscape of datacenters. My research indicates that QLC SSD is a promising contender when comparing against other types of SSDs and HDDs. While QLC has the worst write/read IOPS when compared to other SSDs, it does not detract from the fact that QLC SSD is more suitable for large data workloads compared with small data workloads. QLC SSD can also provide a favorable solution to datacenters with its high capacity and density. In the cost-effective analysis, I concluded that one QLC SSD is equivalence in performance as a 5-7 HDDs RAID array at a similar cost. Also, QLC SSD has lower power consumption while retaining higher reliability than HDD.

The further acceleration experiments of QLC SSD prove that QLC SSD performance is affected by environment temperature well beyond the recommended 25°C, including degradation of I/O throughput and IOPS. However, Latency is rarely affected. Big data benchmark workloads tests also indicate the borderline temperature of major degradation is at 50°C. And, degradation actually start early once temperature increases, the more higher the temperature, the more significant degradation can observe.

All these study of SSD, including the characteristics of SSD, the reliability of SSD, and the performance evaluation of SSD, reveal some important understandings that never known. At the same time, I am not only stressing some common senses of SSD, but proofing them with strict experiments and tests, and supporting them with quantify statistic.

9.2. List of Publications in My Ph.D. Studies

(1) **Shuwen Liang**, Zhi Qiao, Sihai Tang, Jacob Hochstetler, Song Fu, Weisong Shi, Hsing-Bung Chen, "An Empirical Study of Quad-Level Cell (QLC) NAND Flash SSDs for Big Data Applications", In Proceedings of the IEEE International Conference on Big Data (Big Data), December 2019.

(2) **Shuwen Liang**, Zhi Qiao, Jacob Hochstetler, Song Huang, Song Fu, Weisong Shi, Devesh Tiwari, Hsing-Bung Chen, Bradley Settlemyer, David Montoya, "Reliability characterization of solid state drives in a scalable production datacenter", In Proceedings of the IEEE International Conference on Big Data (Big Data), December 2018.

(3) **Shuwen Liang**, Zhi Qiao, Song Fu, Weisong Shi, "In-Depth Reliability Characterization of NAND Flash based Solid State Drives in High Performance Computing Systems", In The 47th International Conference on Parallel Processing (ICPP), August 2018.

(4) Song Huang, **Shuwen Liang**, Song Fu, Weisong Shi, Devesh Tiwari, Hsing-Bung Chen "Characterizing disk health degradation and proactively protecting against disk failures for reliable storage systems", In Proceedings of the IEEE International Conference on Autonomic Computing (ICAC), 2019.

(5) Zhi Qiao, **Shuwen Liang**, Hsing-Bung Chen, Song Fu, Bradley Settlemyer, "Exploring Declustered Software RAID for Enhanced Reliability and Recovery Performance in HPC Storage Systems", In Proceedings of the The 38th Symposium on Reliable Distributed Systems (SRDS), October 2019.

(6) Zhi Qiao, **Shuwen Liang**, Song Fu, Hsing-Bung Chen, Bradley Settlemyer, "Characterizing and Modeling Reliability of Declustered RAID for HPC Storage Systems",

In Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) –Industry Track, June 2019.

(7) Zhi Qiao, **Shuwen Liang**, Nandini Damera, Song Fu, Hsing-bung Chen, Michael Lang, "ACTOR: Active Cloud Storage with Energy-Efficient On-Drive Data Processing", In Proceedings of the IEEE International Conference on Big Data (Big Data), December 2018.

(8) Zhi Qiao, Jacob Hochstetler, **Shuwen Liang**, Song Fu, Hsing-bung Chen, Bradley Settlemyer, "Incorporate Proactive Data Protection in ZFS Towards Reliable Storage Systems", In Proceedings of the IEEE 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DataCom), August 2018.

(9) Zhi Qiao, Jacob Hochstetler, **Shuwen Liang**, Song Fu, Hsing-bung Chen, Bradley Settlemyer, "Developing cost-effective data rescue schemes to tackle disk failures in data centers", In Proceedings of the International Conference on Big Data, June 2018.

(10) Zhi Qiao, **Shuwen Liang**, Hai Jiang, Song Fu, "A customizable MapReduce framework for complex data-intensive workflows on GPUs", In Proceedings of The IEEE 34th International Performance Computing and Communications Conference (IPCCC), December 2015.

(11) Zhi Qiao, **Shuwen Liang**, Hai Jiang, Song Fu, "MR-Graph: a customizable GPU MapReduce", In Proceedings of The IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), November 2015.

9.3. Future Directions

The growing adoption of edge computing and IoT infrastructure shapes the requirements of future storage systems. Indeed, the future storage system will keep a pace to not only have larger capacity, higher density but also more reliable. SSD is one of the promising future by its cost-effective feature, especially the QLC SSD. There is also no doubt that as the 3D NAND technology developing, higher density of bits will be available in one cell in the near future. On the other hand, another trend for future storage media is multi-functionality. Combing memory and storage together, the storage media can achieve higher transportation speed and better performance. OED drive is an example of the early stage of hybrid disk drive.

As a future work, I plan to further study the reliability degradation process of SSDs and accurately model this process for a deeper understanding and better characterization of SSD reliability. From my previous studies, even though significant degradation is observed, but the cause of degradation remains mystery.

My future work will be focus on four different categories. Firstly, SSD can performances other different stress tests, for example, rapid thermal tests or vibration tests. Quantify the degradation and modeling the degradation process will be important to predict the SSD performance and disk failure. Secondly, controller of SSD embedded a lot management algorithms that can affects the performance and reliability of SSD. For example, garbage collection algorithm, wear-leveling algorithm and so forth. Understand these algorithms help me have a much better understanding of the reliability of SSD. And it also can help to modify relative algorithms to optimize performance for specific working scenarios. Thirdly, there are some hybrid SSDs are in the high end storage market. And there is lack of reliability analysis of those SSDs. Testing and evaluate the performance and reliability of hybrid SSD will help us to understand them better. Last by not least, I would like to study and develop the cost-effective data placement algorithms based on my previous studies of the SSD's performance and reliability characteristics.

REFERENCES

[1] *blktrace- linux man page.*

[2] *Data center cooling technologies and best practices.*

[3] *fio - flexible i/o tester rev. 3.27.*

[4] *Halt and hass testing services.*

[5] *Hgst open ethernet drive architecture*, `https://www.openstack.org/summit/openstack-summit-atlanta-2014/session-videos/presentation/demo-theater-hgst-open-ethernet-drive-architecture`.

[6] *Hgst's openethernet drive*, `http://www.hgst.com`.

[7] *Intel-bigdata.*

[8] *Wattsup power meter*, `https://www.wattsupmeters.com`.

[9] *Zfec - a fast erasure codec which can be used with the command-line*, `https://pypi.org/project/zfec/`.

[10] *Validating the reliability of intel® solid-state drives*, 2011, archived.

[11] *Why ethernet switches can take the heat (or cold)*, `https://iebmedia.com/technology/why-ethernet-switches-can-take-the-heat-or-cold/`, 2014.

[12] *3d xpoint*, 2019, https://en.wikipedia.org/wiki/3DXPoint.

[13] *Gray code*, 2019, https://en.wikipedia.org/wiki/Graycode.

[14] *Multi-level cell*, 2019, https://en.wikipedia.org/wiki/Multi-levelcell.

[15] *Nvme*, 2019, https://en.wikipedia.org/wiki/NVM Express.

[16] *Serial ata*, 2019, https://en.wikipedia.org/wiki/SerialATA.

[17] *S.m.a.r.t.*, 2019, https://en.wikipedia.org/wiki/S.M.A.R.T.

[18] *Ssd*, 2019, https://en.wikipedia.org/wiki/Solid-statedrive.

[19] *Storage performance benchmarking with fio*, 2019, https://thesanguy.com/tag/block-size/.

[20] *Trim*, 2019, https://en.wikipedia.org/wiki/Trim(computing).

[21] *How to manage complexity and realize the value of big data*, 2020,

https://www.ibm.com/blogs/services/2020/05/28/how-to-manage-complexity-and-realize-the-value-of-big-data/.

[22] *M.2 nvme*, `https://www.atpinc.com/products/industrial-ssds-nvme-m.2-wide-temperature`, 2020.

[23] *Safe ssd operating temperature: Is it too hot for your ssd to run?*, `https://garageencoderspro.com/safe-ssd-operating-temperature-is-it-too-hot-for-your-ssd-to-run/`, 2020.

[24] *Hadoop vs. spark: What's the difference?*, 2021.

[25] AKCP, *How temperature affects it storage*, `https://www.akcp.com/blog/how-temperature-affects-it-data-storage/`, 2021.

[26] L. S. S. Reddy B. Thirumala Rao, *Scheduling data intensive workloads through virtualization on mapreduce based clouds*, International Journal of Distributed and Parallel Systems (IJDPS), vol. 3, 2012, pp. 99–110.

[27] Lakshmi N Bairavasundaram, Garth R Goodson, Shankar Pasupathy, and Jiri Schindler, *An analysis of latent sector errors in disk drives*, Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2007, pp. 289–300.

[28] BRIAN BEERS, *P-value*, `https://www.investopedia.com/terms/p/p-value.asp`, 2021.

[29] H. P. Belgal, N. Righos, I. Kalastirsky, J. J. Peterson, R. Shiner, and N. Mielke, *A new reliability model for post-cycling charge retention of flash memories*, Proc. 40th Annual (Cat. No.02CH37320) 2002 IEEE Int Reliability Physics Symp, April 2002, pp. 7–20.

[30] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann, *Predicting disk replacement towards reliable data centers*, Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD), Acm, 2016, pp. 39–48.

[31] A. Brand, K. Wu, S. Pan, and D. Chin, *Novel read disturb failure mechanism induced by flash cycling*, Proc. 31st Annual Reliability Physics 1993, March 1993, pp. 127–132.

[32] Hsing bung Chen, *Offloading erasure coding operation from storage server to ethernet disk drive based storage system - a parallel and scalable solution*, Los Alamos national Lab (2016).

[33] Y. Cai, Y. Luo, S. Ghose, and O. Mutlu, *Read disturb errors in mlc NAND flash memory: Characterization, mitigation, and recovery*, Proc. 45th Annual IEEE/IFIP Int. Conf. Dependable Systems and Networks, June 2015, pp. 438–449.

[34] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, *Data retention in mlc NAND flash memory: Characterization, optimization, and recovery*, Proc. IEEE 21st Int. Symp. High Performance Computer Architecture (HPCA), February 2015, pp. 551–563.

[35] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, *Program interference in mlc NAND flash memory: Characterization, modeling, and mitigation*, Proc. IEEE 31st Int. Conf. Computer Design (ICCD), October 2013, pp. 123–130.

[36] Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu, *Errors in flash-memory-based solid-state drives: Analysis, mitigation, and recovery*, CoRR abs/1711.11427 (2017).

[37] Yu Cai, Erich F Haratsch, Onur Mutlu, and Ken Mai, *Error patterns in mlc nand flash memory: Measurement, characterization, and analysis*, Proceedings of the Conference on Design, Automation and Test in Europe, EDA Consortium, 2012, pp. 521–526.

[38] Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F Haratsch, Adrian Cristal, Osman S Unsal, and Ken Mai, *Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime*, Computer Design (ICCD), 2012 IEEE 30th International Conference on, Ieee, 2012, pp. 94–101.

[39] P. Cappelletti, R. Bez, D. Cantarelli, and L. Fratin, *Failure mechanisms of flash cell in program/erase cycling*, Proc. IEEE Int. Electron Devices Meeting, December 1994, pp. 291–294.

[40] P. Carns, K. Harms, W. Allcock, C. Bacon, S. Lang, R. Latham, and R. Ross, *Understanding and improving computational science storage access through continuous char-*

acterization, Proc. IEEE 27th Symp. Mass Storage Systems and Technologies (MSST), May 2011, pp. 1–14.

[41] Feng Chen, David A Koufaty, and Xiaodong Zhang, *Understanding intrinsic characteristics and system implications of flash memory based solid state drives*, ACM SIGMETRICS Performance Evaluation Review, vol. 37, Acm, 2009, pp. 181–192.

[42] H. Chen and S. Fu, *Passi: A parallel, reliable and scalable storage software infrastructure for active storage system and I/O environments*, 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), Dec 2015, pp. 1–8.

[43] H. Chen, G. Grider, J. Inman, Parks, Fields, and J. A. Kuehn, *The impact of vectorization on erasure code computing in cloud storages - a performance and power consumption study*, Proc. IEEE 8th Int. Conf. Cloud Computing, June 2015, pp. 781–788.

[44] R. Degraeve, F. Schuler, B. Kaczer, M. Lorenzini, D. Wellekens, P. Hendrickx, M. van Duuren, G. J. M. Dormans, J. Van Houdt, L. Haspeslagh, G. Groeseneken, and G. Tempel, *Analytical percolation model for predicting anomalous charge loss in flash memories*, IEEE Transactions on Electron Devices 51 (2004), no. 9, 1392–1400.

[45] Ben Eckart, Xin Chen, Xubin He, and Stephen L Scott, *Failure prediction models for proactive fault tolerance within storage systems*, Ieee Mascots, 2008.

[46] Shiri Gaber, Oshry Ben-Harush, and Amihai Savir, *Predicting hdd failures from compound smart attributes*, Proceedings of the 10th ACM International Systems and Storage Conference, 2017, pp. 1–1.

[47] ALEX GLAWION, *How hot is too hot for a gpu? – graphics card temperature guide*, `https://www.cgdirector.com/gpu-temperature-guide/`, 2021.

[48] Jim Gray and Catharine Van Ingen, *Empirical measurements of disk failure rates and error rates*, arXiv preprint cs/0701166 (2007).

[49] Laura M Grupp, John D Davis, and Steven Swanson, *The bleak future of nand flash memory*, Proceedings of the 10th USENIX conference on File and Storage Technologies, USENIX Association, 2012, pp. 2–2.

[50] Haryadi S Gunawi, Riza O Suminto, Russell Sears, Casey Golliher, Swaminathan Sun-

dararaman, Xing Lin, Tim Emami, Weiguang Sheng, Nematollah Bidokhti, Caitie McCaffrey, et al., *Fail-slow at scale: Evidence of hardware performance faults in large production systems*, ACM Transactions on Storage (TOS) 14 (2018), no. 3, 1–26.

[51] Brent Hale, *Safe cpu temps: How hot should my cpu be?*, `https://techguided.com/safe-cpu-temp/`, 2021.

[52] Song Huang, Song Fu, Quan Zhang, and Weisong Shi, *Characterizing disk failures with quantified disk degradation signatures: An early experience*, IEEE International Symposium on Workload Characterization (IISWC), 2015.

[53] Tae-Sung Jung, Young-Joon Choi, Kang-Deog Suh, Byung-Hoon Suh, Jin-Ki Kim, Young-Ho Lim, Yong-Nam Koh, Jong-Wook Park, Ki-Jong Lee, Jung-Hoon Park, Kee-Tae Park, Jang-Rae Kim, Jeong-Hyong Lee, and Hyung-Kyu Lim, *A 3.3 V 128 Mb multi-level NAND flash memory for mass storage applications*, Proc. ISSCC 1996 IEEE Int. Solid-State Circuits Conf.. Digest of TEchnical Papers, February 1996, pp. 32–33.

[54] M. Kato, N. Miyamoto, H. Kume, A. Satoh, T. Adachi, M. Ushiyama, and K. Kimura, *Read-disturb degradation mechanism due to electron trapping in the tunnel oxide for low-voltage flash memories*, Proc. IEEE Int. Electron Devices Meeting, December 1994, pp. 45–48.

[55] Y. Kim, R. Gunasekaran, G. M. Shipman, D. A. Dillow, Zhe Zhang, and B. W. Settlemyer, *Workload characterization of a leadership class storage cluster*, Proc. 5th Petascale Data Storage Workshop (PDSW '10), November 2010, pp. 1–5.

[56] Jae-Duk Lee, Jeong-Hyuk Choi, Donggun Park, and Kinam Kim, *Degradation of tunnel oxide by fn current stress and its effects on data retention characteristics of 90 nm NAND flash memory cells*, Proc. 41st Annual 2003 IEEE Int. Reliability Physics Symp, March 2003, pp. 497–501.

[57] Jing Li, Xinpu Ji, Yuhan Jia, Bingpeng Zhu, Gang Wang, Zhongwei Li, and Xiaoguang Liu, *Hard drive failure prediction using classification and regression trees*, IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2014.

[58] Shijun Liu and Xuecheng Zou, *Qlc nand study and enhanced gray coding methods for sixteen-level-based program algorithms*, Microelectron. J. 66 (2017), no. C, 58–66.

[59] Yixin Luo, *Architectural techniques for improving nand flash memory reliability*, Ph.D. Thesis, School of Computer Science Carnegie Mellon University (2018).

[60] Ao Ma, Rachel Traylor, Fred Douglis, Mark Chamness, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu, *RAIDShield: characterizing, monitoring, and proactively protecting against disk failures*, ACM Transactions on Storage 11 (2015), no. 4, 17.

[61] Farzaneh Mahdisoltani, Ioan Stefanovici, and Bianca Schroeder, *Proactive error prediction to improve storage system reliability*, 2017 USENIX Annual Technical Conference (USENIX ATC 17), 2017, pp. 391–402.

[62] F. Margaglia and A. Brinkmann, *Improving mlc flash performance and endurance with extended p/e cycles*, Proc. 31st Symp. Mass Storage Systems and Technologies (MSST), May 2015, pp. 1–12.

[63] Justin Meza, Qiang Wu, Sanjev Kumar, and Onur Mutlu, *A large-scale study of flash memory failures in the field*, ACM SIGMETRICS Performance Evaluation Review, vol. 43, Acm, 2015, pp. 177–190.

[64] C. Miccoli, J. Barber, C. M. Compagnoni, G. M. Paolucci, J. Kessenich, A. L. Lacaita, A. S. Spinelli, R. J. Koval, and A. Goda, *Resolving discrete emission events: A new perspective for detrapping investigation in NAND flash memories*, Proc. IEEE Int. Reliability Physics Symp. (IRPS), April 2013, pp. 3b.1.1–3b.1.6.

[65] N. Mielke, T. Marquart, Ning Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, *Bit error rate in NAND flash memories*, Proc. IEEE Int. Reliability Physics Symp, April 2008, pp. 9–19.

[66] Joseph F Murray, Gordon F Hughes, and Kenneth Kreutz-Delgado, *Hard drive failure prediction using non-parametric statistical methods*, Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP), 2003.

[67] Iyswarya Narayanan, Di Wang, Myeongjae Jeon, Bikash Sharma, Laura Caulfield,

Anand Sivasubramaniam, Ben Cutler, Jie Liu, Badriddine Khessib, and Kushagra Vaid, *Ssd failures in datacenters: What? when? and why?*, Proceedings of the 9th ACM International on Systems and Storage Conference, Acm, 2016, p. 7.

[68] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso, *Failure trends in a large disk drive population.*, Proceedings of the 8th USENIX Conference on File and Storage Technologies, 2007.

[69] James S Plank, *Erasure codes for storage systems: A brief primer*, Login: The USENIX Magzine (2013).

[70] Zhi Qiao, Song Fu, Hsing-Bung Chen, and Bradley Settlemyer, *Building reliable high-performance storage systems: An empirical and analytical study*, 2019 IEEE International Conference on Cluster Computing (CLUSTER), Ieee, 2019, pp. 1–10.

[71] Zhi Qiao, Jacob Hochstetler, Shuwen Liang, Song Fu, Hsing-bung Chen, and Bradley Settlemyer, *Incorporate proactive data protection in ZFS towards reliable storage systems*, Proceedings of IEEE International Conference on Big Data Intelligence and Computing (DataCom), Ieee, 2018, pp. 904–911.

[72] Carlos AC Rincón, Jehan-François Pâris, Ricardo Vilalta, Albert MK Cheng, and Darrell DE Long, *Disk failure prediction in heterogeneous environments*, 2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Ieee, 2017, pp. 1–7.

[73] Cristian Zambelli Rino Micheloni, Luca Crippa and Piero Olivo, *Architectural and integration options for 3d nand flash memories*, Computers 6(3) (2017), no. 27.

[74] Patrick Schmid, *elociraptor returns: 6gb/s, 600gb, and 10,000 rpm*, 2010.

[75] Bianca Schroeder and Garth A Gibson, *Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you?*, Proceedings of the 8th USENIX Conference on File and Storage Technologies, 2007.

[76] Bianca Schroeder and Garth A. Gibson, *Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?*, 5th USENIX Conference on File and Storage Technologies (FAST 07) (San Jose, CA), USENIX Association, February 2007.

[77] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant, *Flash reliability in production: The expected and the unexpected.*, Fast, 2016, pp. 67–80.

[78] Thomas Schwarz, Mary Baker, Steven Bassi, Bruce Baumgart, Wayne Flagg, Catherine van Ingen, Kobus Joste, Mark Manasse, and Mehul Shah, *Disk failure investigations at the internet archive*, Work-in-Progess session, NASA/IEEE Conference on Mass Storage Systems and Technologies (MSST2006), Citeseer, 2006.

[79] Xuecheng Zou Shijun Liu, *Analysis of 3d nand technologies and comparison between charge-trap-based and floating-gate-based flash devices*, Journal of Universities of Posts and Telecommunications 24 (2017), 75–96.

[80] Peter Desnoyers Simona Boboila, *Write endurance in flash drives: Measurements and analysis*, 2010.

[81] Softing, *Pci controlnet:high-temperature network card for pci bus computers.*

[82] H. Sun, W. Liu, Z. Qiao, S. Fu, and W. Shi, *DStore: A holistic key-value store exploring near-data processing and on-demand scheduling for compaction optimization*, IEEE Access 6 (2018), 61233–61253.

[83] Bo Mao Hong Jiang Suzhen Wu, Yanping Lin, *Gcar: Garbagecollection aware cache management with improved performance forflash-based ssds*, Proceedings of ACM International Conference on Supercomputing (ICS) (2016).

[84] Y. Takai, M. Fukuchi, R. Kinoshita, C. Matsui, and K. Takeuchi, *Analysis on heterogeneous ssd configuration with quadruple-level cell (qlc) nand flash memory*, Proceedings of IEEE International Memory Workshop (IMW), 2019.

[85] K. Takeuchi, S. Satoh, T. Tanaka, K. Imamiya, and K. Sakui, *A negative vth cell architecture for highly scalable, excellently noise immune and highly reliable NAND flash memories*, Proc. Symp. VLSI Circuits. Digest of Technical Papers (Cat. No.98CH36215), June 1998, pp. 234–235.

[86] H. Tseng, L. Grupp, and S. Swanson, *Understanding the impact of power loss on flash memory*, Proceedings of ACM/IEEE Design Automation Conference (DAC), 2011.

[87] Kashi Venkatesh Vishwanath and Nachiappan Nagappan, *Characterizing cloud comput-*

ing hardware reliability, Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 193–204.

[88] Yu Wang, Qiang Miao, Eden WM Ma, Kwok-Leung Tsui, and Michael G Pecht, *Online anomaly detection for hard disk drives based on mahalanobis distance*, IEEE Transactions on Reliability 62 (2013), no. 1, 136–145.

[89] SungJin Whang, KiHong Lee, DaeGyu Shin, BeomYong Kim, MinSoo Kim, JinHo Bin, JiHye Han, SungJun Kim, BoMi Lee, YoungKyun Jung, SungYoon Cho, ChangHee Shin, HyunSeung Yoo, SangMoo Choi, Kwon Hong, Seiichi Aritome, SungKi Park, and SungJoo Hong, *Novel 3-dimensional dual control-gate with surrounding floating-gate (dc-sf) nand flash cell for 1tb file storage application*, International Electron Devices Meeting (2010).

[90] S. Wu, H. Li, B. Mao, X. Chen, and K. Li, *Overcome the gc-induced performance variability in ssd-based raids with request redirection*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38 (2019), no. 5, 822–833.

[91] Jiang Xiao, Zhuang Xiong, Song Wu, Yusheng Yi, Hai Jin, and Kan Hu, *Disk failure prediction in data centers via online learning*, Proceedings of the 47th International Conference on Parallel Processing, 2018, pp. 1–10.

[92] Qin Xin, Thomas JE Schwarz, and Ethan L Miller, *Disk infant mortality in large storage systems*, 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Ieee, 2005, pp. 125–134.

[93] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, et al., *Improving service availability of cloud systems by predicting disk error*, 2018 USENIX Annual Technical Conference (USENIX ATC 18), 2018, pp. 481–494.

[94] R. Yamada, Y. Mori, Y. Okuyama, J. Yugami, T. Nishimoto, and H. Kume, *Analysis of detrap current due to oxide traps to improve flash memory retention*, Proc. 38th Annual (Cat. No.00CH37059) 2000 IEEE Int Reliability Physics Symp, April 2000, pp. 200–204.

[95] R. Yamada, T. Sekiguchi, Y. Okuyama, J. Yugami, and H. Kume, *A novel analysis*

method of threshold voltage shift due to detrap in a multi-level flash memory, Proc. Symp. VLSI Technology. Digest of Technical Papers (IEEE Cat. No.01 CH37184), June 2001, pp. 115–116.

[96] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng, *Predicting disk failures with hmm-and hsmm-based approaches.*, Springer ICDM, 2010.

[97] Mai Zheng, Joseph Tucek, Feng Qin, and Mark Lillibridge, *Understanding the robustness of ssds under power fault*, Proceedings of USENIX Conference on File and Storage Technologies (FAST), 2013.

[98] Qi Zhu, Xiang Li, and Yinan Wu, *Thermal managerment of high power memory module for server platforms*, 2008 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 2008, pp. 572–576.