# ENHANCED 3-TIER STORAGE MANAGEMENT SCHEME FOR FLASH MEMORY-BASED SOLID STATE DISK

AHMED I. N. ALSALIBI

UNIVERSITI SAINS MALAYSIA

2017

# ENHANCED 3-TIER STORAGE MANAGEMENT SCHEME FOR FLASH MEMORY-BASED SOLID STATE DISK

**by**

**AHMED I. N. ALSALIBI**

**Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy**

**July 2017**

# DEDICATION

To my Father, may Allah bless his soul, forgive him of his sins, make his grave a garden and grant him the highest levels of paradise.

# ACKNOWLEDGEMENT

First of all, I am grateful to Allah for the good health and wellbeing that were necessary to complete this research.

I would like to express my special appreciation and thanks to my supervisor Associate Professor Putra Sumari, you have been a tremendous mentor for me. I would like to thank you for encouraging conducting my research and for allowing me to grow as a research scientist. Your advice on research as well as on my career has been priceless.

I wish to express my sincere thanks to USM and School of Computer Sciences for providing me all the necessary facilities for the research. I place on record my sincere thank to School of Computer Sciences for the financial support in exchange for my work as graduate assistant during my study. I am also grateful to the lecturers, with whom I have worked in the School of Computer Sciences. I am extremely thankful and indebted to them for sharing expertise, sincere and valuable guidance and encouragement extended to me.

I take this opportunity to thank the Malaysian people for their continued hospitality and showing us a brilliant time in their beautiful country.

Special thanks to my family. Words cannot express how grateful I am to my mother for all the sacrifices that you have made for me. My sincere thanks also go to my brothers (Rami, Wesam, and Mohammed) and my sisters (Rania, Lena, Soha, and Heba). Your prayers for me were what sustained me hitherto. I would also like to thank my special friend Mohammad Shehab, who supported me, and encouraged me to strive for my goal. Special thank to Dr. Mohammad Shambour, I am deeply indebted to his friendship, which helped me whenever I felt frustrated.

At the end, I would like to express appreciation to my beloved wife Rana and my son Izzat, who have spent a long time without my presence at home. Rana was always my support in the moments when there was no one to answer my queries.

# TABLE OF CONTENTS

**CHAPTER 1 – INTRODUCTION**

**CHAPTER 2 – LITERATURE REVIEW**

**CHAPTER 3 – RESEARCH METHODOLOGY**

## CHAPTER 4 – CLUSTERING-BASED GARBAGE COLLECTION SCHEME (CGC)

## CHAPTER 5 – THREE-TIER FLASH MEMORY-BASED STORAGE MANAGEMENT SCHEME (TM) FOR HYBRID SSD

CHAPTER 6 – CONCLUSION

# LIST OF TABLES

# LIST OF FIGURES

**Page**

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AFVM | Asymmetric Flash Volume Management |
| ASU | Application Specific Unit |
| BAST | Block Associative Sector Translation |
| CAT | Cost Age Time |
| CATA | Cost Age Time with Age Sort |
| CB | Cost Benefit |
| CGC | Clustering-based Garbage Collection (CGC) |
| DABC-NV | Device-Aware Buffer Cache with NVRAM and VRAM |
| DFTL | Demand-based Flash Translation Layer |
| EC | Energy Consumption |
| ECC | Error Correction Code |
| FAST | Fully Associative Sector Translation |
| FeGC | Fast and endurance Garbage Collection |
| FeRAM | Ferroelectric RAM |
| FTL | Flash Translation Layer |
| GC | Garbage Collection |
| GR | Greedy |

| | |
|---|---|
| HDD | Hard Disk Drive |
| TM | Three-tier Flash-based Storage Management Scheme |
| HySSD | Hybrid SSD |
| HYSTOR | Hybrid Storage Solution |
| LPNs | Logical Page Numbers |
| LRU | Least Recent Use |
| MLC | Multi Level Cell |
| MRU | Most Recent Use |
| NVRAM | Non-Volatile RAM |
| OOB | Out Of Band |
| PcRAM | Phase-change RAM |
| PERAL | Performance, Energy, and Reliability Balanced Dynamic Data |
| PPNs | Physical Page Numbers |
| RAM | Random Access Memory |
| ReRAM | Resistive Random-Access Memory |
| RR-FDCA | Round-Robin Frozen Data Collection Algorithm |
| SLC | Single-Level Cell |
| SRAM | Static Random-Access Memory |
| SSD | Solid State Disk |

| | |
|---|---|
| SSHD | Solid State Hybrid Drives |
| TLC | Triple-Level Cell |
| USB | Universal Serial Bus |
| VRAM | Volatile RAM |
| WL | Wear Leveling |
| XIP | Execute In Place |

# PENINGKATAN SKIM 3-PERINGKAT PENGURUSAN STORAN UNTUK MEMORI KILAT – BERASASKAN PEMACU KEADAAN PEJAL

## ABSTRAK

Pada masa kini, banyak tumpuan terarah kepada memori kilat berasaskan pemacu keadaan pejal. Berbeza dengan pemacu tradisional, pemacu keadaan pejal menggunakan cip semikonduktor untuk menyimpan data. Struktur ini merupakan ciri-ciri teknikal asli termasuk penggunaan kuasa yang rendah, rintangan kejutan dan prestasi tinggi dalam capaian rawak. Disebabkan ciri-ciri ini, banyak peranti menggunakan memori flash sebagai komponen simpanan asas, seperti mp3, telefon pintar dan peranti tablet. Walau bagaimanapun, memori kilat, unit asas pemacu keadaan pejal, mempunyai banyak ciri-ciri tersendiri, yang membawa kepada cabaran yang pelbagai sebagai contoh, menulis data (menyimpan data) hanya dibenarkan pada unit simpanan kosong (blok), yang memerlukan masa yang lama. Selain itu, setiap blok mempunyai bilangan kitaran padam yang terhad. Dalam kajian ini, dua skim baru telah dicadangkan. Skim pertama dipanggil CGC (*Pengumpulan Sampah berasaskan Pengklusteran*) yang bertujuan meningkatkan kebolehpercayaan dan prestasi pemacu keadaan pejal. Manakala, yang kedua dipanggil TM (*Skim Tiga-Peringkat Pengurusan Storan berasaskan Memori Kilat*) untuk hibrid pemacu keadaan pejal (iaitu pemacu keadaan pejal yang terdiri daripada jenis memori kilat yang berbeza) untuk mengeksploitasi keseimbangan antara prestasi, kebolehpercayaan, ketumpatan dan penggunaan kuasa. Kelayakan skim CGC dan TM telah dibuktikan dengan menggunakan penyelaku DiskSim. Keputusan menunjukkan bahawa skim-skim yang dicadangkan adalah sangat kompetitif dalam pelbagai beban kerja sebenar. Dari segi kebolehpercayaan, peratusan peningkatan memihak kepada CGC ke atas skim yang lain sebanyak 86.56%. Dari segi prestasi, peratusan peningkatan adalah memihak kepada TM ke atas skim lain berjumlah 185.55%.

# ENHANCED 3-TIER STORAGE MANAGEMENT SCHEME FOR FLASH MEMORY-BASED SOLID STATE DISK

## ABSTRACT

Nowadays, significant attention has been paid to the flash memory-based Solid State Disk (SSD). Different from traditional disks, SSD uses semiconductor chips for storing the data. This structure enjoys original technical characteristics including low power consumption, shock resistance and high performance in random access. Owing to these features, many devices are using flash memory as a basic storage component, such as mp3, smartphones, and tablet devices. However, the flash memory, basic unit of SSD, has many distinctive characteristics, which lead to multifarious challenges for example, writing data (storing data) is only allowed on empty storage unit (block), which makes it more time-consuming. Moreover, each block has limited number of erase cycles. In this research, two novel schemes are proposed. The first one is called Clustering-based Garbage Collection (CGC) aimed at increasing the reliability and performance of SSD. Whereas, the second one is called Three-tier Flash Memory-based Storage Management scheme (TM) for hybrid SSD (i.e. SSD consisting of different types of flash memory) in order to increase the performance, reliability, density, and reduce power consumption. The eligibility of the CGC and TM schemes are proven by using DiskSim simulator. The results reveal that the proposed schemes are very competitive to state-of-the-art schemes in various real-workloads. In terms of reliability, the percentage of enhancement is in the favor of CGC over the other schemes amounts to 86.56 %. In terms of performance, the percentage of enhancement is in the favor of TM over the other schemes amounts to 185.55 % .

# CHAPTER 1

# INTRODUCTION

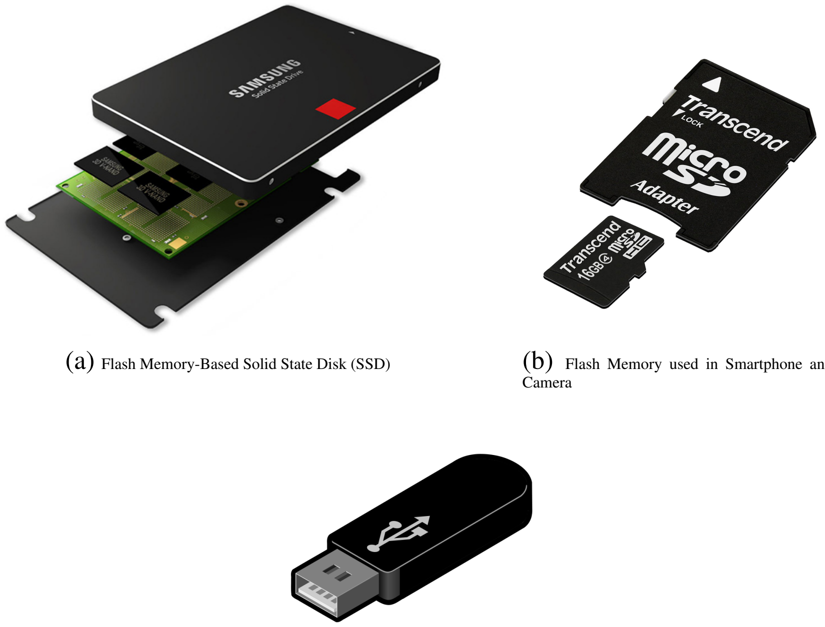
## 1.1 Introduction

Flash memory-based Solid State Disk (SSD) is the most popular storage device being used nowadays. Undoubtedly, it plays vital role in today's storage technology, as it offers countless advantages over the traditional mechanical disk, known as Hard Disk Drive (HDD) (Kang and Jeong, 2015).

SSD uses semiconductor chips (i.e. flash memory) for storing data instead of magnetic platters. This architecture produces better technical features, which include low power consumption, shock resistance, high performance in random access, high density, noiseless, portable, and non-vulnerable to the magnetism's effect (Micheloni et al., 2012; Kang and Jeong, 2015; Kim et al., 2009).

Owing to these features, flash memory has become a pivotal technology today in data storage systems (Chen et al., 2009). Nowadays, many devices are using flash memory as a basic storage component, such as mp3, PC cards, pen-drives, laptops, smartphones, and tablet devices (Helm et al., 2014). However, in the future, the traditional storage devices, such as HDD, will be expectedly replaced by SSD technology. Figure 1.1 shows a few examples of flash memory products.

In accordance with data collected by Trend Focus, as shown in Figure 1.2, the industry sold a total of 30.777 million SSDs during first quarter (Q1) of 2016, increasing



(a) Flash Memory-Based Solid State Disk (SSD)

(b) Flash Memory used in Smartphone and Camera

(c) Flash Memory Drive

Figure 1.1: Flash Memory Products

the sale by 32% as compared to sale of 23.190 million during the same period of 2015 (TrendFocus, 2016).

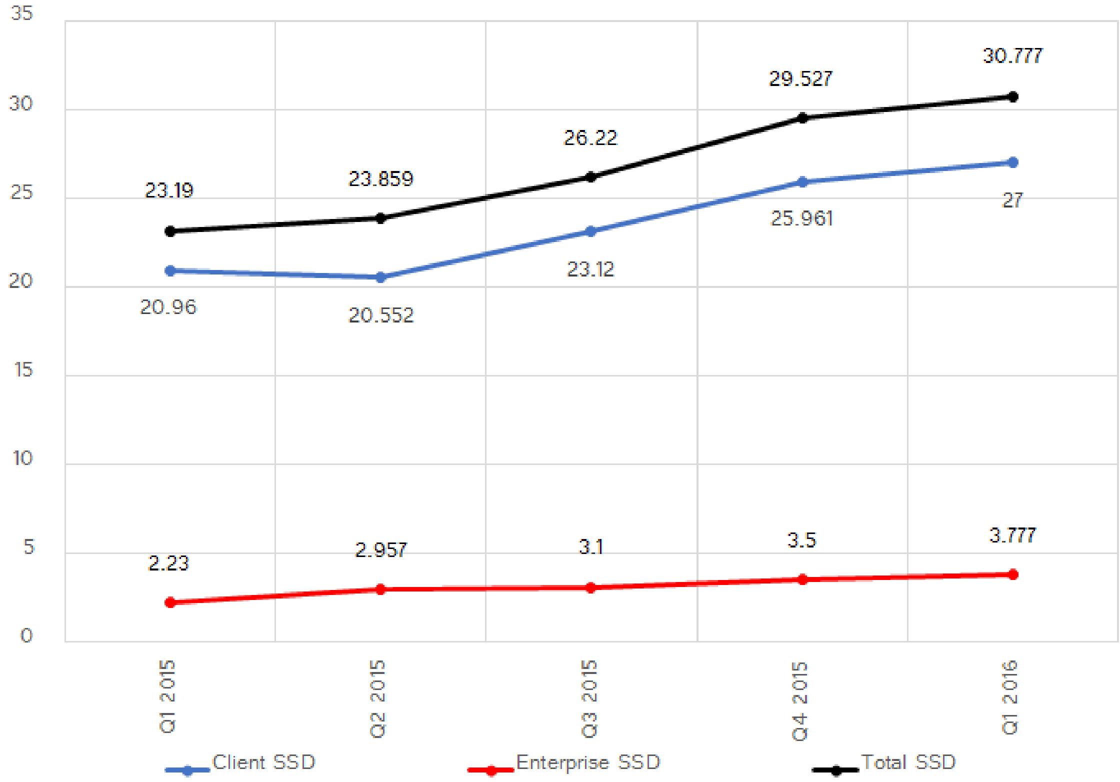


Figure 1.2: Shipments of SSDs for Different Applications (TrendFocus, 2016)

Samsung is the world's largest maker of flash memory and is also the largest manufacturer of SSDs. The company has been controlling over 40% of the market, and its unit shipments increased from 9.42 million in Q1 of 2015 to 12.93 million in Q1 of 2016. Samsung supplies SSDs to large PC makers like Apple, HP and Lenovo (TrendFocus, 2016).

Despite its strength, SSDs, in general, are much expensive than traditional storage devices (Mao et al., 2014). The price gap between SSD and these devices will remain high, which is not expected to be disclosed at least in the near future (Liu et al., 2013).

## 1.2 SSD Performance Metrics

SSD performance is measured by means of various metrics, such as performance speed, reliability, cost, density and power consumption. It is primordial to demonstrate definitions of the factors mentioned above and embrace the most adopted definitions by most researchers, which are concluded as follow:

- Performance speed: Refers to how well the SSD functions while accessing, retrieving or saving data. It also reflects how fast a software application loads and runs and how quickly files are accessed or stored (Dirik and Jacob, 2009).
- Reliability: Refers to lifespan of the SSD, which reflects its durability (i.e. number of erase cycles for each block inside SSD) (Yang et al., 2015).
- Cost: Refers to the price of SSD (i.e. GB/$), cost of design, material and other various charges spent throughout the production process (Min et al., 2012).
- Density: Refers to the SSD capacity, its storage capabilities and the amount of

space available for data loading (i.e. Number of bits per cell) (Hachiya et al., 2014).

- Power consumption: Refers to the consumption of energy or power during the SSD operations (Tiwari et al., 2013).

## 1.3 Problem Statement

As a novel storage technology, SSDs provide many features and challenges. The physical semiconductor characteristics of SSD result into high performance, power consumption, light size, shock resistance, and low noise (Micheloni et al., 2012). Pure SSD (i.e. SSD solely consisting of one type of flash memory either SLC, MLC or TLC) is used in many consumer devices such as tablets, laptops, and personal computers. For the time being, there has been a great interest in embedding this technology in the hierarchy of large-scale storage systems such as servers and data centers in order to improve the I/O operations performance, and to increase the applications' runtime (Tan et al., 2014). With all the benefits of SSD, replacing the conventional HDD drives with pure SSDs may not be a good option for large-scale storage systems (Mao et al., 2014). This is mainly due to many disadvantages of SSD such as limited lifespan, small density, and high cost. Thus, a more practical solution is to use hybrid SSD (i.e SSD consisting of SLC, MLC, and TLC) in hierarchy of storage system, such that the features of this technology are best utilized (Batni and Safaei, 2014). The key challenges are to decide what role should each flash memory has in the storage hierarchy, and what data should be stored in it. In order to build a useful hybrid SSD, this research aims to address two problems as follows:

### 1.3.1 Managing Garbage Collection Operations

SSD architecture consists of blocks, where each one contains a number of pages. Writing data (storing data) is only allowed on empty (free) pages of the block (Chen et al., 2009). Once the page contains data, overwriting on it (updating data) is not allowed (Liu et al., 2013). To do the overwriting, it is necessary to first erase the existing data on those pages, and then overwrite new data on it (Chang et al., 2013).

Furthermore, the erase operation in SSD is allowed only at block level. Erasing on single page is not allowed. Therefore, to erase single page, all the block pages should be erased (Chang et al., 2013). All other pages of that block are erased automatically. Other rule of SSD is that it is manufactured with certain number of erases allowable on each block (called erase cycles) (Liu et al., 2013). Once reach the limit, the block become dead (unusable forever). The erase cycles are associated to SSD lifespan. Small number of erase cycles means short lifespan. These characteristics of SSD have led to the following issues:

(i) When updating is done very frequently, this leads to perform several erasing operations. Erasing activity on blocks needs to be carefully managed, as the SSD blocks have limited number of erase cycles. Hence, if it is reach the limit too soon, it will shorten the lifespan of the SSD. In other words, when data erasing is frequently carried out on the same block, it reduces the lifespan of the block compared to other block witnessing less erasing operations. It is necessary that the erasing operation is managed efficiently among blocks, so that the lifespan can be lasting. (ii) The other issue when updating is done frequently is that it is lead to creating many invalid pages. Invalid page is referred to the page that contains old data and it cannot be used for writing (storing) new data (Kim et al., 2009). This has lead to reduction in number of free blocks inside SSD. The invalid data, called “garbage data”, must be reclaimed and then erased (change to free). Selected number of blocks are chosen for erasing operation and the selection procedure is called Garbage Collection (GC) (Sun et al., 2014). These selected blocks may contain valid and invalid data; thus, before GC mechanism, entire valid data must be moved into the available free space in other free blocks. Selecting blocks (victim blocks) by GC randomly degrades the overall performance of SSD, such as selecting the block with number of valid data more than number of invalid data is increased the over head of moving the valid data to other free blocks. Therefore, the time of GC operation is increased and the performance of SSD is affected. On the other hand, selecting the block that has high number of erase cycles is undesirable and could degrade the reliability of SSD.

The first problem statement is regarding the difficulty of managing the GC operation among SSD blocks in order to enhance the performance and prolong the lifespan of the SSD. One way to do this is to decrease and distribute the number of erase cycles as evenly as possible among all blocks inside SSD.

### 1.3.2 Compensating Trade-off between SSD Flash Memories

Nowadays, there are three types of flash memory, which are available in the market. Depending on the integrated predetermined erase cycles allowable, these three types of flash memory have different characteristics. These three varieties of flash memory (SLC, MLC and TLC) have their own strengths and weakness, as discussed below.

1. Single-Level Cell (SLC): SLC provides high performance and longer lifespan,

whereas it is considerably costly and has a low density (i.e. small storage space).

2. Multiple-Level Cell (MLC): MLC promotes high density and less costly, while its performance and lifespan are weak and short respectively.

3. Triple- Level Cell (TLC): TLC provides high density with very less cost, whereas it has considerably low performance and lifespan (Hsieh et al., 2015; Hachiya et al., 2014).

To achieve the high requirements for high performance storage systems with less cost, a cost-effective solution should be developed by integrating the features of the existing flash memories (SLC, MLC, and TLC) (Oh and Lee, 2013; Chang, 2008; Jimenez et al., 2015). The second problem statement is to integrate the features of these three kinds of flash memory within the same device, which is the main challenge, wherein the trade-off between performance, reliability, density and power consumption should be highly utilized.

The problems addressed in this research can be summarized as follows: given a novel scheme to overcome the dilemma of GC in pure SSD. Thereafter, take the benefit of this scheme to propose a hybrid SSD composed of three partitions: SLC as the primary storage to serve the hot data, MLC and TLC as the secondary storage to serve the warm and cold data respectively. Design a novel scheme to allocate the data in each partition of hybrid SSD, such that it maximizes the data organization that consequently improves the overall performance and reliability of hybrid SSD.

## 1.4 Objectives

The main objectives of this research are:

1. To propose a novel scheme for SSD, which is able to manage the garbage collection operation, in order to increase the performance and reliability of SSD.
2. To propose a novel scheme to manage a hybrid SSD consisting of SLC, MLC, and TLC flash memory in order to increase the performance, reliability, density and reduce power consumption.
3. To adapt and extend the DiskSim simulator in order to evaluate the hybrid SSD.

## 1.5 Research Contributions

The key contributions of this research are:

1. Introducing a new scheme, known as Clustering-based Garbage Collection (CGC) scheme, in order to enhance the performance and reliability of SSD. The performance and reliability issues are enhanced by clustering the data inside SSD to valid and invalid. Valid data is gathered to the first cluster, while the invalid data is gathered to second cluster. This organization of data aims to overcome the dilemma of GC. In case of scarcity of free blocks inside SSD, the garbage collector is invoked. Instead of selecting the victim block from the entire SSD, the most suitable block is selected from cluster 1 or cluster 2 based on the concern of SSD.
2. Come out with a novel scheme, known as Three-tier Flash Memory-based Stor-

age Management scheme (TM) for Hybrid SSD consisting of SLC, MLC and TLC flash memory, in order to increase performance, reliability, density and reduce power consumption. The performance and reliability is increased by serving the heavy workload and exactly the hot data in SLC partition and considering it as a first layer in the hierarchy of SSD. The cost and the density are decreased and increased respectively by serving the warm data in MLC partition and considering it as a second layer in the hierarchy of SSD. Furthermore, the density is increased by serving the cold data in TLC partition and considering it as a third layer in the hierarchy of SSD. Since the cost of TLC type is less than SLC and MLC, thus adding this kind to the hierarchy of SSD aims to shrink the cost of adding SLC and MLC partitions.

## 1.6 Scope of the Research

This research focuses on a specific types of flash memory, called NAND flash memory with SLC, MLC, and TLC, owing to the popularity of these kinds of flash memory, which are widely used in consumer electronics, such as mp3, PC cards, compact flash, laptops, smartphones, and tablet devices. The eligibility of the proposed schemes is proven by means of widely used simulation tools in the literature: DiskSim simulator (Bucy et al., 2008) with SSD extension from Microsoft (Prabhakaran and Wobber, 2009).

## 1.7 Organization of the Thesis

The remaining parts of this thesis are organized as follows. Chapter 2 provides extensive details about state-of-the-art schemes. This chapter initially describes the current storage system approaches in section 2.2. Classification and overview of the previous schemes are presents in section 2.3. The flash memory schemes based SSD are described in section 2.4. The flash memory schemes based hybrid SSD consisting of SLC, MLC, and TLC are described in section 2.5. Finally, critical analysis about the state-of-the-art schemes discussed in section 2.6.

Chapter 3 explains the research methodology and design. Initially, the design of the research methodology is described in section 3.2. Thereafter, the methods of CGC and TM schemes are described in sections 3.3 and 3.4, respectively. Finally, simulation's setup is discussed with further details in section 3.5 including: simulator, simulator design and settings, and benchmark workloads.

Chapter 4 explains the design of the first proposed scheme (CGC). Section 4.2 presents the overview and the details of the proposed scheme and section 4.3 discusses the results.

Chapter 5 explains the design of the second proposed scheme (TM). Section 5.2 outlines the key observations that is lead to propose a new scheme. Section 5.3 presents the overall architecture of the proposed scheme including the details of its components. Finally, section 5.4 discusses the results.

Chapter 6 provides an extensive but not exhaustive conclusion with details about the future directions.

Finally in Appendices, Chapter A provides background about SSD including the architectural design of existing SSD devices in section A.2, Garbage Collection (GC) in section A.3, Wear Leveling (WL) in section A.4, and extensive details about the features of the most popular types of flash memory (i.e. SLC, MLC and TLC) in section A.5. Chapter B provides further implementation and evaluation for CGC scheme using EagleTree simulator.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

This chapter provides extensive details about state-of-the-art schemes. Initially, the current storage system approaches are described in section 2.2. Classification and overview of the previous schemes are presented in section 2.3. The flash memory schemes based SSD are described in section 2.4. The flash memory schemes based hybrid SSD consisting of SLC, MLC, and TLC are described in section 2.5. Critical analysis about the state-of-the-art schemes is discussed in section 2.6. Finally, section 2.7 presents the chapter summary.

## 2.2 Current Storage System Approaches

As a novel storage technology, SSDs provide efficient features and challenges. The physical semiconductor characteristics of SSD result into high performance, power consumption, light size, shock resistance, and low noise (Micheloni et al., 2012). Pure SSD (i.e. SSD solely consisting of one type of flash memory either SLC, MLC or TLC) is used in many consumer devices such as tablets, laptops, and personal computers. For the time being, there has been a great interest in embedding this technology in the hierarchy of large-scale storage systems such as servers and data centers in order to improve the I/O operations performance, and to increase the applications' runtime (Tan et al., 2014). With all the benefits of SSD, replacing the conventional HDD drives with pure SSDs may not be an efficient option for large-scale storage systems, because of the small density and high cost of SSD (Mao et al., 2014). Thus, a more practical solution is to hybridize SSD with HDD (Li et al., 2015).

This solution is integration of SSD and HDD to create new hybrid storage devices with promising features: performance from SSD and the capacity and cheap price from HDD (Seagate, 2014). Thus, many companies try to take advantages of both SSD and HDD to creates new storage device (hybrid drive) by exploring SSD as the primary storage to take the benefits of performance, capacity and cost. For example, (1) PEARL: Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays (Xie and Sun, 2008), (2) Hystor: A Hybrid Storage Solution (Chen et al., 2011), (3) HybridStore: A Cost-Efficient, High-Performance Storage System Combining SSDs and HDDs (Kim et al., 2011), and (4) Hybrid Aggregates: Combining SSDs and HDDs in A single Storage Pool (Strunk, 2012). Moreover, there are also several tiering hybrid drives in industry: (1) Dell a Compellent Flash Array (Dell, 2011), (2) Western Digital's SSHD (Computerworld, 2013), (3) Apple's Fusion Drive (Shimpi, 2012), and (4) Microsoft's Ready Drive (Microsoft, 2014).

Although, the combining of SSD and HDD in the same device can decrease the cost and increase the capacity, managing of both SSD and HDD is very complicated because each device has its own physical characteristics e.g. HDD is mechanical device, whereas SSD is solely consists of integrated circuits. As a result, a hybrid SSD (i.e. SSD consisting of SLC, MLC, and TLC) is proposed and added to hierarchy of storage system, such that the features of this technology are best utilized (Batni and Safaei, 2014). The key challenges are to decide what role should each flash memory have in the storage hierarchy, and what data should be stored in it.

These three kinds of flash memory (i.e. SLC, MLC and TLC) are available in the market. Each kind of flash memory has its own pros and cons. SLC provides some pros (high performance and high lifespan), and some cons (high cost and low density i.e. one bit per cell). MLC provides some pros (high density i.e. two bits per cell and low cost), and some cons (low performance and short lifespan). TLC provides some pros (very high density i.e. three bits per cell and very low cost), and some cons (very low performance and very low lifespan) (Hachiya et al., 2014). To achieve the high requirements for high performance storage systems with low cost, a cost-efficient solution should be developed by combining different kinds of flash memory inside SSD such as SLC, MLC, and TLC (Sung and Kim, 2012). Combining these three kinds of flash memory inside the same device is the main challenge to exploit the trade-off between performance, reliability, cost, density and power consumption.

## 2.3 Classification and Overview

The SSD schemes are divided into two types as shown in taxonomy diagram Figure 2.1 : (a) Flash memory based SSD schemes: schemes that have been proposed to manage SSD with one kind of flash memory and (b) Flash memory based hybrid SSD schemes: schemes that have been proposed to manage hybrid SSD (SSD with different kinds of flash memory SLC, MLC, and TLC).

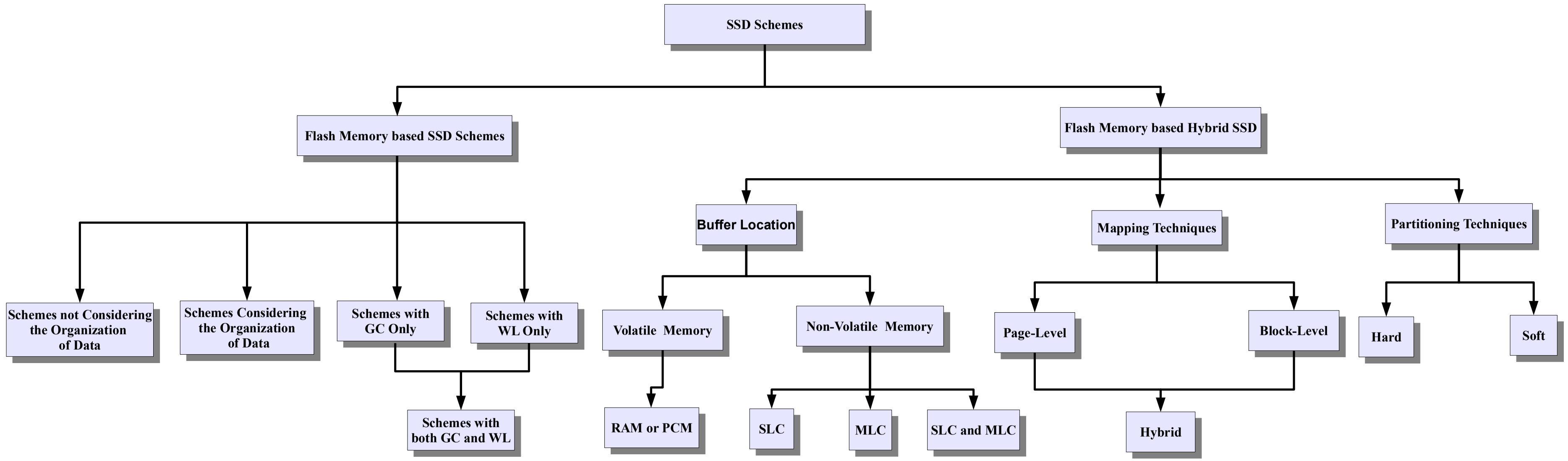


Figure 2.1: Organizations of SSD Schemes

### 2.3.1 Flash Memory based SSD Schemes

The flash memory based SSD schemes could be categorized into three types based on the following architecture designs as shown in taxonomy diagram Figure 2.1: (a) The consideration of GC and WL, which is divided into three categories (*i*) Schemes just considering the GC, (*ii*) Schemes just considering the WL , and (*iii*) Schemes considering both GC and WL.

(b) The allocation policy used in each scheme, which is divided into two categories: (*i*) Schemes using "FIFO" as a block allocation policy, (*ii*) Schemes using "Youngest block first" as a block allocation policy. (c) The organization of hot and cold data inside SSD, which is divided into two types: (*i*) Schemes consider the organization of data , (*ii*) Schemes do not consider the organization of data.

Table 2.1 classifies the previous schemes based on theses architecture design. Hot data refers to data that frequently updated by the users, whereas cold data refers to data that seldom updated by the users.

### 2.3.2 Flash Memory based Hybrid SSD

The flash memory based hybrid SSD schemes could be categorized into three types based on the following architecture designs as shown in taxonomy diagram Figure 2.1: (a) The mapping techniques, (b) Partitioning techniques, and (c) Buffer location.

Table 2.1: Classification Based on the Consideration of GC and WL, the Allocation Policy, and the Organization of Hot and Cold Data

| Category | References |
|---|---|
| The Consideration of GC and WL | |
| Considering the GC | (Wu and Zwaenepoel, 1994; Kawaguchi et al., 1995; Menon and Stockmeyer, 1998) |
| Considering the WL | (Chang, 2007) |
| Considering both GC and WL | (Chiang and Chang, 1999; Kwon et al., 2011; Kim and Lee, 2002; Han et al., 2006) |
| The Allocation Policy | |
| FIFO | (Wu and Zwaenepoel, 1994; Kawaguchi et al., 1995; Chang, 2007; Kim and Lee, 2002) |
| Youngest Block First | (Chiang and Chang, 1999; Menon and Stockmeyer, 1998; Kwon et al., 2011) |
| The Organization of Hot and Cold Data | |
| Considering the Organization of Data | (Chiang and Chang, 1999; Kwon et al., 2011; Han et al., 2006; Kim and Lee, 2002) |
| Not Considering the Organization of Data | (Kawaguchi et al., 1995; Wu and Zwaenepoel, 1994; Menon and Stockmeyer, 1998) |

#### 2.3.2(a) Based on the Mapping Techniques

The previous schemes can be categorized into three different types based on the mapping techniques used to manage either SLC or MLC flash memory as shown in taxonomy diagram Figure 2.1: page-level, block-level, and hybrid-level (Chung et al., 2009; Ma et al., 2014).

Address translation software that embedded inside FTL has responsibility for creating the information inside mapping table between Logical Page Address (LPA) and Physical Page Address (PPA) in the file system and SSD respectively (Grupp et al., 2009; Lee et al., 2011). Whenever a page is updated, new data are always stored in a new location with a new address, whereas the data that reside in the previous location is marked as invalid. Thus, the address of pages and blocks should be continuously updated to keep the track of addressing between file system and flash memory.

The major challenge is to achieve high performance mapping without consuming too much space inside the SRAM. That is because SRAM is a highly expensive memory in the market due to its high performance. It is normally embedded within the SSD architecture to retain the mapping address translation table. Accordingly, the designer of SRAM tends to minimize its size (Ma et al., 2014). Table 2.2 classifies the previous schemes based on the mapping techniques used in hybrid SSD design.

In *page-level* mapping technique, a logical page can be directly mapped to any location inside the flash memory. By means of this, a high flexibility to separate hot and cold data is retained (Oh and Lee, 2013; Im and Shin, 2010). However, page-level requires a large size of address translation table, because it should store all addresses

Table 2.2: Classification Based on Mapping Techniques

| Category | References |
|---|---|
| Page-Level | (Im and Shin, 2010; Oh and Lee, 2013) |
| Block-Level | (Chang, 2010) |
| Page-Level and Block Level | (Im and Shin, 2009; Lu et al., 2012; Murugan and Du, 2012) |
| Hybrid-Level | (Jung and Song, 2009; Nam et al., 2010; Jimenez et al., 2012, 2013) |

for each page inside flash memory, thus, consuming the size of SRAM.

In *block-level* mapping technique, the process of mapping operations in the address translation table is performed at the block level instead of page level. In this technique, the size of the address translation table requirements is less than the requirements of address translation table size in page-level technique which is obviously equal to the number of blocks inside the flash memory, thus the size of SRAM partition required is smaller. However, the hot and cold data are hardly separated. In the previous architectural design of flash memory software schemes, there are many research efforts that considered the block-level technique in FTL software schemes (Chang, 2010).

In *hybrid-level* mapping technique, both page-level and block-level techniques are hybridized to complement their advantages. In the hybrid technique, a block-mapping table is used to extract the physical block address from the logical block address. Then, after reaching the selected physical block, the location of the page inside the block is determined from the inner page-mapping table, which is located in the selected block (Jung and Song, 2009; Nam et al., 2010; Jimenez et al., 2012, 2013).

#### 2.3.2(b) Based on the Partitioning Techniques

The previous schemes can be also categorized based on the partitioning types: hard and soft as shown in taxonomy diagram Figure 2.1.

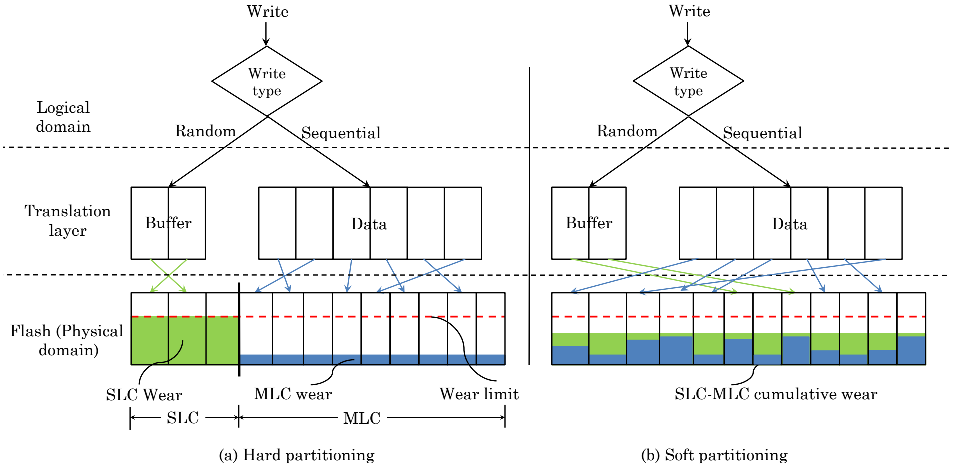


Figure 2.2: (a) Hard Partitioning (SLC and MLC chips are physically separated. Mapping of random writes to SLC can lead to its early wear-out, degrading overall lifetime) (b) Soft Partitioning (Only MLC blocks are used, some of which can be selectively programmed as SLC to improve performance and achieve wear-leveling) (Jimenez et al., 2012)

In hard partitioning, SLC and MLC chips are physically separated and a particular chip continues to work as SLC or MLC during entire execution time as shown in Figure 2.2. There are many software schemes that have been proposed using hard partitioning technique such as (Yim, 2005; Jung and Song, 2009; Im and Shin, 2009, 2010; Chang, 2008; Murugan and Du, 2012; Park et al., 2012; Lu et al., 2012; Park et al., 2011; Nam et al., 2010).

In Soft partitioning,the MLC blocks can be selectively woks as SLC blocks which keeps performance close to that of SLC. There are many schemes that have been proposed using soft partitioning technique such as (Sung and Kim, 2012; Jimenez et al.,

Table 2.3: Classification Based on Partitioning Techniques

| Category | References |
|---|---|
| Hard Partitioning | (Yim, 2005; Jung and Song, 2009; Im and Shin, 2009, 2010; Chang, 2008; Murugan and Du, 2012; Park et al., 2012; Lu et al., 2012; Park et al., 2011; Nam et al., 2010) |
| Soft Partitioning | (Sung and Kim, 2012; Jimenez et al., 2012, 2013; Lee and Kim, 2014) |

2012, 2013; Lee and Kim, 2014). Table 2.3 classifies the previous schemes based on the partitioning techniques used in hybrid SSD design.

#### 2.3.2(c) Based on the Buffer Location

The previous schemes can be also categorized into five types based on the buffer location located in the hierarchy of each scheme as shown in taxonomy diagram Figure 2.1:

(*i*) Schemes using SLC as a buffer (SLC-Buf), the main advantage of this type is increasing the performance of SSD in terms of speed. However, the shortcoming of such scheme is the use of a small size of SLC as a buffer to serve the hot data, thus the SLC partition is worn-out quickly and that affects the overall reliability of hybrid SSD such as (Chang, 2008; Jung and Song, 2009; Im and Shin, 2009, 2010; Nam et al., 2010; Jimenez et al., 2012, 2013).

(*ii*) Schemes using MLC as a buffer (MLC-Buf), the main advantage of this type is increasing the density and decreasing the cost of hybrid SSD. However, same to SLC-Buf schemes the shortcoming of these schemes is the use of a small size of MLC as a buffer to serve the hot data, thus the MLC partition is wear-out very quickly and that is seriously affect the overall reliability of SSD such as (Hachiya et al., 2014).

(*iii*) Schemes using both SLC and MLC as a buffer (SLC/MLC-Buf), unlike the previous schemes which considers the SLC as a buffer, these schemes combine both SLC partition and MLC partition as a buffer. The main advantages of these schemes are: (a) Increasing the reliability of SSD, (b) Having more flexibility, and (c) Cost efficient (Lee and Kim, 2014).

(*iv*) Schemes using external memory as a buffer (ExTM-Buf), instead of using either SLC or MLC as a buffer, these schemes utilize especial types of memory as a buffer such as RAM, PRAM, and DRAM. The main advantage of these schemes is the ability of bridging the disparity among the storage system and main memory in terms of speed. Conversely, the high price of these kinds of memory and the possibility of lost the data in case of the power outage are the main challenges to consider in the architecture of enterprise storage system such as (Yim, 2005; Park et al., 2011; Lu et al., 2012; Park et al., 2012).

(*v*) Schemes without a Buffer (No-Buf), these schemes do not consider the buffer in their hierarchy, which degrades the performance of hybrid SSD in terms of speed and increase the cost of the SSD because they used a large size of SLC in their hierarchy as compared to the previous schemes such as (Sung and Kim, 2012; Murugan and Du, 2012). Table 2.4 classifies the previous schemes based on hybrid SSD architecture and buffer location.

Table 2.4: Classification Based on Hybrid SSD Architecture and Buffer Location

| Category | References |
|---|---|
| Hybrid SSD Architecture | |
| MLC+TLC | (Hachiya et al., 2014) |
| SLC+MLC+TLC | (Oh and Lee, 2013) |
| SLC+MLC | Nearly all others |
| Memory/Cell Used as Buffer | |
| SLC | (Chang, 2008; Jung and Song, 2009; Im and Shin, 2009, 2010; Nam et al., 2010; Jimenez et al., 2012, 2013) |
| MLC | (Hachiya et al., 2014) |
| SLC + MLC | (Lee and Kim, 2014) |
| External Memory | (Yim, 2005; Park et al., 2011; Lu et al., 2012; Park et al., 2012) |
| No Buffer | (Sung and Kim, 2012; Murugan and Du, 2012) |

## 2.4 Garbage Collection Schemes

As the SSD does not support in-place-updating, thus FTL was designed to forward the write operations to blank blocks inside SSD (Gupta et al., 2009). Reserved amount of blank blocks should be always available to receive the upcoming write operations. In case there are consumption of free blocks, the GC is invoked by FTL in order to create free space for upcoming write operations.

In order to design an efficient GC software, several questions have to be answered (Chiang and Chang, 1999): (*i*) When must the GC occur and start? (*ii*) Which block must be erased? and (*iii*) How must the GC take place? During the GC operations, many blocks are selected for the erase operation. Then the filtering process is carried out as follows: Blocks with solely invalid data is selected first. Note that this process does not need any complex operation and it perform without degrading the speed of SSD.

On other hand, blocks with mixed valid and invalid data must go through other efficient selection procedure. Obviously, selecting the blocks with a number of invalid data more than valid data is better than selecting the block with valid data more than invalid data. Since each block has a particular lifespan, efficient scheme is required to guarantee that the blocks are not become invalid before they are properly utilized. The main responsibility of the block selection procedure is to reduce the erase operation cost and to preserve the blocks from reaching the lifespan quickly.

The GC procedure operates as follows: (*i*) selects a victim block among invalid and valid blocks based on different criteria such as number of erase cycle of each block, number of invalid data in each block, time since last update, etc. (*ii*) once a valid block is selected, transfers valid pages from the victim block to another free block, (*iii*) erases the victim block, (*iv*) adds the erased block to the free block list (Kim and Kwak, 2016).

### 2.4.1 Why Block Selection Scheme is Important?

The operation of the GC has to be run repeatedly to create free space. This process consumes a huge number of write and erase operations; efficient victim block selection is very important in preventing the degradation of SSD performance and improving the speed of I/O. Figure 2.3 illustrates the process of GC and demonstrates the significance of the procedure for victim block selection. Three free blocks and six valid and invalid blocks are considered in this example. GC is triggered to create more free blocks. Selecting the victim blocks is accomplished through three scenarios (Kwon et al., 2011).

In Scenario 1, blocks 2 and 4 are selected as victim blocks by the garbage collector. Accordingly, the valid pages are transferred from these blocks to other free blocks, and