RB-Explorer: An Accurate and Practical Approach to Write Amplification Measurement for SSDs

Hui Sun, Xiao Qin, *Senior Member, IEEE*, Hong Jiang, *Senior Member, IEEE*,
Jianzhong Huang, and Changsheng Xie, *Member, IEEE*

Abstract—A large write amplification ratio degrades the program/erase cycles (P/Es) of NAND Flashes and reduces the endurance and performance of solid state disks (SSDs). The lack of a practical way to measure write amplification for SSDs motivates us to propose a novel measuring method called RB-Explorer at the SSD level rather than the NAND Flash level. The goal of RB-Explorer is two-fold: (1) to accurately measure the write amplification of SSDs to quantify SSD endurance and (2) to study the impacts of I/O techniques on write amplification of SSDs. RB-Explorer incorporates a Ready/Busy (R/B) signal of one of the NAND Flashes in an SSD in a proposed write amplification model for SSDs with four full-parallelism levels (i.e., the channel, chip, die, and plane levels). RB-Explorer takes two steps toward measuring write amplification. First, RB-Explorer quantifies the number of page programs using the low R/B signal level, the duration of which varies with the different operation (i.e., read, program, and erase) in NAND Flash. Second, RB-Explorer measures data volume written to NAND Flashes by considering parallelisms at four levels. Data volume written to a die in a NAND Flash is obtained as a product of the number $\mathrm{N_p}$ of programs and page size $\mathrm{P_a}$. Given the number $\mathrm{N_{channel}}$ of channels, the number $\mathrm{N_{chip}}$ of chips per channel, and the number $\mathrm{N_{die}}$ of dies per chip, one can obtain the data volume written to NAND Flashes as a product of $\mathrm{N_p}, \mathrm{P_a}, \mathrm{N_{die}}, \mathrm{N_{chip}}$, and $\mathrm{N_{channel}}$. RB-Explorer is applied to analyzing write amplification ratios of SSDs to track SSD endurance. Furthermore, we implement a real-world SSD (i.e., SSD-v) and employ a fine-tuned SSD simulator (i.e., SSDsim) to validate the accuracy of RB-Explorer. Our experimental results show that RB-Explorer improves on the accuracy of SSDsim—the state-of-the-art SSD simulator—in most tested cases. We conduct a series of measurements using micro-benchmarks and I/O traces to demonstrate how RB-Explorer may be applied to investigate SSDs.

Index Terms—Write amplification, NAND flash, solid state disk, read/busy signal, write endurance, performance

1 Introduction

Write amplification has strong impacts on the endurance and performance of solid state disks (SSDs). This paper presents a novel and practical method called RB-Explorer to measure write amplification at the SSD level rather than the NAND Flash level. At the heart of RB-Explorer is a model of write amplification for SSDs with four full parallelism levels (i.e., the channel, chip, die, and plane levels). Our RB-Explorer can be applied to track SSD endurance by analyzing the write amplification of SSDs. We implement a real-world SSD (i.e., SSD-v) and a fine-tuned SSD simulator (i.e., SSDsim) to validate the accuracy and credibility of our RB-Explorer. Micro-benchmarks and I/O traces are loaded on tested SSDs to evaluate the write amplification of the tested SSDs.

SSD endurance is limited by the number of program/erase cycles (P/Es) in a NAND Flash can endure before being worn out. Due to out-of-place and erase-before-write updates in NAND Flash, an updated page must be rewritten to an available page in another block or the same block after erasing its enclosing block. These two processes introduce excessive P/Es, this is, amplify page programs, which reduces the lifetime of SSDs. Increasing the utilization of available P/Es can substantially improve SSD lifetime.

More precisely, write amplification is the ratio of data volume written in NAND Flash by the SSD controller to data volume written by the host machine of the SSD. High write amplification indicates a large number of excessive page programs that reduce available P/Es and degrade SSD endurance. Although existing studies quantitatively evaluate the impacts of affecting factors (e.g., over-provisioning (OP) and garbage collection) [3], [12], [13], [14], [15], [16], [22] on write amplification, such quantitative studies were conducted using probabilistic models implemented in simulators. The simulators, which have not been validated by real-world measurements, typically ignore critical SSD device features like parallelisms at the channel, chip, die, and plane layers.

Extensive studies on SSD write amplification have been conducted from the SSD simulators points of view [3], [4], [7], [43]. However, we find in experiments (see Section 4.4) that state-of-the-art SSD simulators face an accuracy challenge in write amplification measurement. The lack of a

- *H. Sun, J. Huang, and C. Xie are with National Laboratory for Optoelectronics, School of Computer Science and Technology, Huazhong University of Science and Technology, 430074, Wuhan, China.*
E-mail: sunhuiworking@gmail.com, hjzh@hust.edu.cn, cs_xie@hust.edu.cn.
- *X. Qin is with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849. E-mail: xqin@auburn.edu.*
- *H. Jiang is with the Department of Computer Science & Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588. E-mail: jiang@cse.unl.edu.*

Manuscript received 8 Aug. 2013; accepted 29 Dec. 2013. Date of publication 25 Feb. 2014; date of current version 13 Mar. 2015.
Recommended for acceptance by E.-Y. Chung.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2014.2308207

practical and accurate way to measure write amplification for SSDs possess a challenge to researchers who design and optimize SSDs because most of such designs and optimizations aim to increase SSD performance and endurance that are directly and significantly affected by write amplification. This challenge motivates us to propose RB-Explorer that is designed (1) to accurately estimate the write amplification of SSDs to quantify SSD endurance and (2) to study the impacts of I/O techniques on write amplification of SSDs.

A centerpiece of RE-Explorer is a write amplification model, the underpinning of which is our key observation that the number of page programs can be quantified by the number of occurrences of the low R/B signal level for page program operations. Our proposed model relies on an observation that the duration of the low level of the R/B signal varies with different operations (i.e., read, program, and erase) in NAND Flash. RB-Explorer measures the R/B signal of one of the NAND Flashes in an SSD by applying the White-Box test [25]—a tested SSD enclosure is opened so that the output level of an R/B pin in one of the NAND Flashes can be tested.

Considering the parallelisms at the four levels inside an SSD, our method is capable of measuring data volume written to NAND Flashes by SSD controllers. Data volume written to a die in a NAND Flash is obtained as a product of the number N_p of programs and page size P_a. Given the number $N_{channel}$ of channels, the number N_{chip} of chips per channel, and the number N_{die} of dies per chip, one can measure the data volume written to NAND Flashes as a product of $N_{channel}$, N_{chip}, N_{die}, N_p, and P_a (see (7) in Section 3.3).

RB-Explorer reported in this paper can be viewed as a step towards accurately measuring write amplification for an entire SSD rather than NAND Flashes. Write amplification for an entire SSD is based on (1) the data volume written from a host and (2) the data volume written to NAND Flashes. Note that the former data volume is determined by I/O workloads and can be technically assessed at the user level; the latter one can be evaluated by our new method.

RB-Explorer has the following four salient features that facilitate the optimization of system performance and reliability:

- First, RB-Explorer enables users to quantify SSD endurance under various workload conditions.
- Second, RB-Explorer offers ample opportunities for system designers to understand end-to-end implications of optimization strategies to boost SSD endurance. For example, one may use RB-Explorer to study the side effects of I/O schedulers and file systems on write amplification.
- Third, RB-Explorer provides guidelines for hardware and software developers to investigate excellent techniques that are focused on reducing write amplification to improve the endurance of SSDs.
- Last but not the least, RB-Explorer can be applied to evaluating the performance and reliability impacts of high-level techniques (e.g., file systems) and low-level techniques (e.g., compression techniques) on the endurance of SSDs in the system. Thus, RB-Explorer allows system designers to deploy SSDs with small write amplification into a system, thereby improving the system's data storage reliability and availability.

This paper makes three main contributions:

- *A new write amplification measurement approach:* Our RB-Explorer not only evaluates SSD performance (e.g., MB/s and IOPS), but also measures data volume written in NAND Flashes of SSDs. RB-Explorer incorporates a four-level model of write amplification for SSDs. The four levels considered in our model include the channel, chip, die, and plane levels. The model in RB-Explorer makes use of an R/B signal of one of the NAND Flashes in an SSD. To improve RB-Explorer, we apply the white-box testing, in which a tested SSD enclosure is opened to measure R/B signals in the NAND Flash to obtain write amplification values.
- *A newly implemented SSD (SSD-v) system:* We implement an SSD system called SSD-v to verify the accuracy and credibility of our RB-Explorer. Our verification results show that the $|RE_{WA\_RB}|$ value, i.e., the relative error of write amplification in Section 4.3, is smaller than 1 percent in 100 percent-write micro-benchmarks and $|RE_{WA\_RB}|$ is smaller than 10 percent in mixed-write micro-benchmarks. The relative error of our RB-Explorer is much smaller than that of a fine-tuned SSD simulator, SSDsim, as shown in Section 4.4. This SSD-v, along with results of SSDsim, shows that RB-Explorer is accurate and the measurement technique is generally applicable to SSDs.
- *Measuring the impact of SSD write amplification:* We conduct a series of measurements using micro-benchmarks and I/O traces to study the impact of write amplification on SSDs. We investigate the *relationship between write amplification and data volume written by a host* (or *WALVD* for short).

The rest of this paper is organized as follows: Background and motivation are provided in Section 2. Section 3 presents the design of RB-Explorer. The *SSD-v* and RB-Explorer validation are presented in Section 4. Section 5 describes experimental results. In Section 6, we survey the related work. The summary of our study and future work can be found in Section 7.

2 Background and Motivation

2.1 NAND Flash and Solid State Disks

NAND Flash [6] is comprised of one or more targets, each of which is organized into one or more dies. A die is the atomic unit that can independently execute commands and report statuses by an R/B signal. Each die is comprised of one or more planes, each of which contains many blocks. Each block contains a fixed number of pages. NAND Flash can execute three different operations, i.e., read, program (write), and erase. Like a block for being the atomic unit for erase operations, a page is the atomic unit for read and program. Within a block, pages are read sequentially or randomly; however, program operations can only access a page sequentially. If any pages need to be updated, only out-of-place and erase-before-write updates are allowed. An excessive number of updates increase write amplification and degrade the available P/Es in NAND Flash, thereby shortening SSD lifetime.

An SSD is mainly comprised of NAND Flashes, cache, and an SSD controller. Cache improves performance of small writes and temporarily stores a mapping table. The SSD controller is the most important component and contains an intermediate software layer called flash translation layer (i.e., FTL). FTL mainly performs address mapping, garbage collection, wear-leveling, etc.

Because out-of-place and erase-before-write updates degrade P/Es and reduce NAND Flash endurance, new hardware and software in SSDs are designed to extend NAND Flash endurance and prolong SSD lifetime.

2.2 Quantifying Write Amplification

There are two ways of defining write amplification. *The first one* is based on NAND Flash, whereas *the second one* investigated in this study, is based on an entire SSD.

Write amplification is a ratio of the actual number of page programs per user page program [3]. This definition assumes that I pages in a *block_A* with $(V + I)$ valid pages have been updated. Thus, the I pages are set to be invalid after the new content of these pages are written elsewhere (out-of-place update). The V pages should be rewritten to another free block when *block_A* is reclaimed by garbage collection. In other words, there are $(V + I)$ pages being rewritten during the updating of I pages. Consequently, the write amplification (*WA*) is given as

$$WA = \frac{V + I}{I}. \tag{1}$$

To measure write amplification values under this definition, the structure and codes about page programs in firmware of an SSD must be given. It is impractical, if not impossible, for anyone other than the SSD's designers to figure out these features of the page programs.

The second definition [11] calculates write amplification as a ratio of data volume written to the NAND Flashes by an SSD controller to data volume written from a host. In Fig. 1 (see Section 3.1), there are two kinds of data volume in the data stream from the host to NAND Flashes. *The data volume written from the host under a certain workload* must be stored in the SSD; this is called *logical data volume* (i.e., *L_Volume_Data*). When the SSD controller writes the total logical data to the NAND Flashes, *the data volume written to the NAND Flashes* is called *physical data volume* (i.e., *P_Volume_Data*). Given these definitions, write amplification is estimated as

$$WA = \frac{P\_Volume\_Data}{L\_Volume\_Data}. \tag{2}$$

The physical data volume is no less than that of the logical one, meaning that the value of write amplification is no less than 1. Two types of data volume should be measured to calculate the write amplification. The *logical data volume* can be obtained by users; the *physical data volume* can be measured practically and accurately by RB-Explorer which will be described in Section 3.

2.3 Two Use Cases of RB-Explorer

There is a rich set of scenarios in which our RB-Explorer can provide SSD designers and system developers with an effective means of measuring write amplification of SSDs. The following two use cases show the importance of our approach:

- An increasing number of new I/O schedulers and file systems are emerging to boost the performance and reliability of SSD-based storage systems. These advanced techniques may have side effects on the write amplification of SSDs. RB-Explorer enables system developers of these I/O techniques to accurately quantify write amplification. Furthermore, RB-Explorer can help system developers understand end-to-end implications of optimization strategies to boost SSD endurance.
- Thanks to SSDs' lower energy consumption and higher I/O performance, SSDs are widely adopted as storage devices for HPC systems such as supercomputers (e.g., the 'Gordon' Supercomputer [26]). The reliability of such supercomputers depends on the endurance of SSDs. To choose the most appropriate SSDs for an SSD-based supercomputer, one may collect traces from the supercomputer and replay real-world traces on candidate SSDs. Applying our RB-Explorer, the supercomputer's designers can choose the best SSD among all candidate SSDs by measuring their write amplification values. In doing so, the system designers are able to optimize the performance of the supercomputer for target applications running on the system.

3 Design of RB-Explorer

In this study, we adopt and improve the *White-Box test* to evaluate write amplification, meaning that we should open an SSD's enclosure to test an R/B signal of one NAND Flash. The duration of a low level of R/B varies with different operations in NAND Flash. In our RB-Explorer, we scan the output level of an R/B signal to calculate the number of the low levels of the R/B signal for page programs. This process can help us obtain the amount of page programs under a given workload condition. Because the page size in a given type of NAND Flash is constant, data volume written by a controller to an NAND Flash can be obtained as the product of the number of page programs and the page size of the NAND Flash. Thanks to the parallelism nature inside an SSD, the data volume written to all NAND Flashes can be obtained. It is a novel method to measure the write amplification value of SSDs under I/O workloads. In what follows, we explain the details of the new and practical RB-Explorer.

3.1 Ready/Busy (R/B) Signal in a NAND Flash

Fig. 1 shows the architecture of a typical SSD. R/B signals, one of which is contained in one die, indicate the status of dies in NAND Flash. A low-level R/B signal indicates that an operation command in the die is in progress.

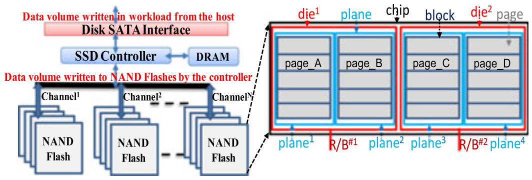


Fig. 1. Typical SSD architecture. In the above NAND Flash of the SSD architecture, there are two dies (i.e., die^1 and die^2) in the chip. Each die consists of two planes (i.e., plane^1 and plane^2 in die^1; plane^3 and plane^4 in die^2). Each plane is composed of blocks, one of which has a fixed number of pages. Four pages (i.e., page_A, page_B, page_C, and page_D) are programmed in four planes (i.e., plane^1, plane^2, plane^3, and plane^4), respectively.

An R/B pin of NAND Flash is an open-drain, active-low output [10], by which we can observe the completion of read, program, and erase operations. The signal is typically at high level for no operations and switches to low level when any one starts. The different low-level durations of R/B signals represent distinct activities in NAND Flash. Because the timing diagrams of R/B for read, program, or erase are similar, we only present the timing diagram of program in Fig. 2.

Fig. 2 depicts the process of the basic page program. First, it requires loading the 80h command (i.e., serial data input) into the command register, followed by five address cycles, and data. The 10h command (i.e., program) is written after the data-input. Then, the page program begins, and the R/B signal stays low for $T_{program}$, which is the duration of the page program time. When page program is complete, the level of R/B returns to the high level. Because $T_{program}$ is different for pages from low address to high address in NAND Flash (especially in MLC NAND Flash), the value of $T_{program}$ is anywhere between a and b μs rather than a constant. In Fig. 2, it means that one $T_{program}$ (i.e., $T_{program} \in (a, b)\mu s$) in one R/B signal contained indicates one page program in a plane. There is also a multi-plane command, where one $T_{program}$ indicates two page program operations in two planes (see, for example, plane^1 and plane^2 of die^1 in Fig. 1). The addresses of the two pages programmed in two planes of one die must be identical.

An R/B signal indicates operations in one die. According to the parallelism among dies, chips, and channels [17], [19], [20], [24], the operations in one die can imply operations in other dies including those on different chips.

3.2 Parallelism and Program Models of SSDs

Parallelism improves SSD performance at four different levels, namely, channel-level, chip-level, die-level, and plane-level (see Fig. 1). The first three parallelism levels are applied to the SSD architecture. Sometimes, plane-level parallelism may not be applied in the single-plane page program model. There are two popular program models in NAND Flash, namely, *the single-plane* and *the multi-plane*[1] page program models.

3.2.1 Full-Parallelism Mode

There is a lack of a universal parallelism model characterizing all the available SSDs on the market. A write amplification model for one parallelism mode might not be applicable for SSDs supporting another parallelism mode. In this study, we focus on modeling write amplification of SSDs with four full-parallelism levels (i.e., channels-level, chips-level, dies-level, and planes-level), because a vast majority of SSDs share such a full-parallelism mode. The prevalent full-parallelism mode aims to boost performance and wear-leveling efficiency in SSDs by performing operations across multiple channels. Sample popular SSDs that support the full-parallelism mode include, but not limited to the Intel 310 series, the Micron Real SSD P400e series, the Intel 520 series, and the OCZ Octane series SSDs, etc.

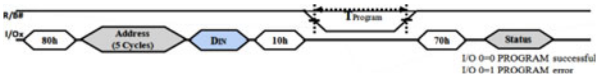


Fig. 2. The timing diagram of a basic page program operation.

SSDs with the non-full-parallelism mode are not widely adopted in the IT industry; and consequently, we do not address the modeling issue of the non-full-parallelism-based SSDs. Nevertheless, our model can be extended to evaluate write amplification of SSDs with the non-full-parallelism mode.

Our full-parallelism model can be viewed as a case study demonstrating how to apply the RB-Explorer approach to measure write amplification of SSDs. If an SSD's parallelism is different (e.g., non-parallelism mode) from the one described in our manuscript, then we will follow the RB-Explorer approach to build a new write amplification model for the SSD's new parallelism mode. In other words, RB-Explorer can be applied to any SSD regardless of the parallelism mode adopted in the SSD.

3.2.2 Non-Hybrid Page Program Model

In this study, we focus on SSDs that either support single-plane page program model or multi-plane page program model (i.e., non-hybrid page program). It does not imply by any means that our page program model cannot be extended to deal with the hybrid page program (i.e., some NAND Flashes are characterized as the single-plane page program model whereas others are expressed as the multi-plane page program model).

It is worth noting that the hybrid page program has not been widely adopted in the industry due to the performance and cost issues. First, the multi-plane page program is superior to the single-plane page program in performance. Given an SSD supporting the hybrid page program, multi-plane-based NAND Flashes in the SSD are slowed down by single-plane-based NAND Flashes residing in the same SSD. In addition, the hybrid page program is unable to take the full advantage of planes-level parallelism. Second, unlike the non-hybrid page program, the hybrid page program relies on an extra hardware component in its SSD controller to determine the type of page program model for the SSD's NAND Flashes. Such a complicated SSD controller increases the cost of hybrid-page-program-enabled SSDs without yielding any performance benefits.

The aforementioned performance and cost issues pertinent to the hybrid page program inspire us to incorporate the non-hybrid page program in our SSD model.

[1] The most popular form of multi-plane in modern NAND Flashes is two-plane (i.e., there are two planes in a die).

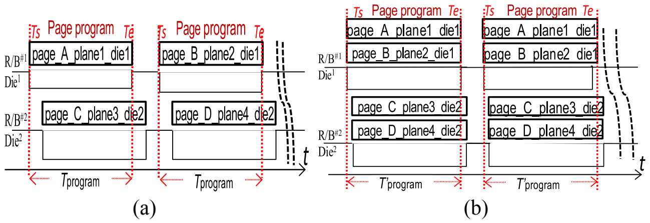


Fig. 3. Two page program models: (a) Single-plane page program model, (b) Multi-plane page program model. In both models, dies are interleaved with one another.

3.2.3 Single-Plane and Multi-Plane Page Program Model

The block diagram of a typical NAND Flash can be found in Fig. 1. In the first program model (see Fig. 3a), two pages (page_A in plane^1 and page_B in plane^2 or page_C in plane^3 and page_D in plane^4) in two respective planes of a die perform the single-plane page program operation. $T_{program}$, the duration of an R/B signal (R/B^1 in die^1 or R/B^2 in die^2) being low for page program, can reflect the time of the page program in a plane. Thus, the single-plane page program model incurs two changes of the R/B signal in a die (die^1 or die^2); the two dies in the same chip execute the interleaved operation using the mechanism of dies-level parallelism. The addresses of the two pages in two planes of one die are not restricted.

In the second model (see Fig. 3b), $T_{program}$ of an R/B signal in a chip reflects two page program operations in parallel between two planes of the same die (page_A in plane^1 and page_B in plane^2 of die^1 or page_C in plane^3 and page_D in plane^4 of die^2). It is restricted in that the addresses of the two pages programmed in the respective two planes of one die must be identical. Two dies in the same chip execute interleaved operations.

According to the parallelism among chips (see Fig. 1), multiple chips on a channel run interleaved operations to guarantee the interleaved operations among dies that are contained in the chips on one channel. It is deemed that an operation indicated from one R/B signal contained in one die is simultaneously executed in all dies of chips belonging to the same channel.

The channel-level parallelism enables the chips on multiple channels to execute, in a parallel fashion, the same operation. Operations among chips on one channel are identical to ones on any of the channels.

By the parallelism levels, it is possible for an operation indicated from one R/B signal contained in one die to be simultaneously running in all dies of chips on any of the channels (or all dies of SSD). In other words, the operation in one die is the same as the ones in other dies, regardless of whether the dies are in the same chip or not. The full parallelism always exploits on page programs even when the request size is smaller than a page. This feature enables us to measure the page program operations based on an R/B signal in any one NAND Flash within an SSD; the corresponding R/B signal is sufficient to conclude that all of the other NAND Flashes are active.

3.3 Write Amplification Model

Now we present our write amplification model driven by the four-level parallelisms (i.e., channel level, chip level, die level, and plane level) of an SSD. The model makes use of an R/B signal in a NAND Flash to measure write amplification. For convenience, we summarize the notations used throughout this paper in Table 1.

TABLE 1
The Notation Used Throughout This Paper

Notation	Description
WA	Value of write amplification for an SSD
L_Volume_Data	Data volume written from a host under workload
P_Volume_Data	Data volume written to NAND Flash by a controller
$N_{channel}$	The number of channels in an SSD
N_{chip}	The number of chips per channel
N_{die}	The number of dies per chip
N_{plane}	The number of planes per die
M_p	The number of parallel planes in a die
P_a	Page size in an NAND Flash of an SSD
R/B	Ready/Busy pin in NAND Flash
T_s ***and*** T_e	Start and end time for slave-recorder system
$T_{program}$	Duration of a page program
(a, b)	The interval of $T_{program}$ (i.e., *Tprogram* ∊ *(a, b)*
WA_{true}	The true value of write amplification
$WA_{measured}$	The measured value of write amplification, including $WA_{measured\_RB}$ in RB-Explorer; $WA_{measured\_SIM}$ in SSDsim
N_{page_true}	The true number of page programs
$N_{page\_measured}$	The measured number of page programs, including $N_{page\_measured\_RB}$ by RB-Explorer; $N_{page\_measured\_SIM}$ in SSDsim
$\lvert RE_{WA}\rvert$	The relative error of WA, including $\lvert RE_{WA\_RB}\rvert$ in RB-Explorer; $\lvert RE_{WA\_SIM}\rvert$ in SSDsim
F_{SIM_RB}	The ratio of $\lvert RE_{WA\_SIM}\rvert$ to $\lvert RE_{WA\_RB}\rvert$
OP%	The percentage of over-provisioning (OP) in the available space of an SSD
available space	Physical space in an SSD
user_Space	The space visible to user in an SSD

From (2), the values of *L_Volume_Data* and *P_Volume_Data* must be given *a priori* to derive the value of write amplification. *L_Volume_Data* depends on workload and *P_Volume_Data* is a product of the numbers of page programs and the page size. The size of a page is constant for a particular NAND Flash; the number of page programs can be easily obtained. Using the parallelism in an SSD and the number of $T_{program}$ in one die for programs, one can determine the number of programs for the entire SSD, in which the value of *P_Volume_Data* can be calculated. Next, write amplification of an SSD under a workload condition can be measured.

In Fig. 1, there are $N_{channel}$ channels, N_{chip} chips per channel, N_{die} dies per chip, N_{plane} planes per die and the size of a page in the NAND Flash is P_a. A die contains one R/B pin, which is selected to count the $T_{program}$ value. The page program duration in a tested NAND Flash is an interval between a and b μs. The total number $N_{Tprogram}$ of $T_{program}$ in one die under a workload condition should be recorded when page programs are fully completed.

Because there are two page program models in NAND Flash (see Section 3.2), the number of pages programmed during the time period of $T_{program}$ is different in these two models. For *the single-plane page program model*, one page is programmed during each $T_{program}$ in a die. And multiple pages are programmed during each $T_{program}$ in a die in *the multi-plane page program model*.

According to the die-level parallelism, the numbers of page programs in two dies of one chip are approximately equal, meaning that $T_{program}$ in each R/B of two dies is the same during the page program. The number of page programs is equal to each other between two dies of the same chip. The same relationship holds true for any two dies of the same NAND Flash in an SSD (see Fig. 1) based on the chips-level and channels-level parallelism. The concurrency of $N_{channel}$ channels in the SSD ensures this relationship between any two dies whether they are contained in the same NAND Flash or not. The number of $T_{program}$ based on an R/B signal can be used to measure the number of page programs in one die under a workload. The same relationship in dies of all NAND Flashes in the SSD allows us to calculate the total number of page programs on the SSD under the workload.

Given a workload condition, the write amplification model derives the total number of page programs using the R/B signal of a die in one NAND Flash. Thus, the model counts the number $N_{Tprogram}$ of $T_{program}$ and obtains the number M_p of parallel planes in a die. The total number N_p of page programs during each $T_{program}$, can be computed as

$$N_p = N_{Tprogram} \times M_p, \tag{3}$$

where the value of M_p depends on the page program model (see Section 3.2.3) in the tested NAND Flash. As shown in (4), M_p is 1 for the single-plane page program model; M_p equals to N_{plane} when it comes to the multi-plane page program model. An approach to quantifying M_p is detailed in Section 3.5

$$M_p = \begin{cases} 1 & \text{single-plane page program model,} \\ N_{plane} & \text{multi-plane concurrent page program model.} \end{cases} \tag{4}$$

Let P_a denote the size of one page; we can calculate the data volume written by an SSD controller to one die as (i.e., $(N_p \times P_a)$). For the NAND Flashes in the SSD, there is one R/B pin per die. Given a chip consisting of N_{die} dies, the number of R/B pins in the chip is N_{die}.

The number of R/B pins in a channel is a product of N_{die} and the number N_{chip} of NAND Flashes in the channel. In other words, the number of R/B pins in the channel is $(N_{chip} \times N_{die})$. Given the parallelism of dies, chips, and channels, multiple (i.e., $(N_{channel} \times N_{chip} \times N_{die})$) dies are identical. Hence, the data volume, *P_Volume_Data*, written to all the NAND Flashes of the SSD is a product of $N_{channel}, N_{chip}, N_{die}, N_p$, and P_a. Thus, we have

$$P\_Volume\_Data = N_{channel} \times N_{chip} \times N_{die} \times N_p \times P_a. \tag{5}$$

It follows from (3) that

$$\begin{aligned} P\_Volume\_Data = N_{channel} \times N_{chip} \times N_{die} \\ \times \left(N_{Tprogram} \times M_p\right) \times P_a. \end{aligned} \tag{6}$$

Using *L_Volume_Data*, P_a, $N_{Tprogram}$, and $T_{program}$, we quantify the write amplification (see also (2)) of an SSD as

$$\begin{aligned} WA &= \frac{P\_Volume\_Data}{L\_Volume\_Data} \\ &= \frac{N_{channel} \times N_{chip} \times N_{die} \times P_a \times (N_{T_{program}} \times M_p)}{L\_Volume\_Data}. \end{aligned} \tag{7}$$

Recall that *L_Volume_Data* in (7) is the data volume written from a host to the SSD under specific workload. *L_Volume_Data* can be measured in the form of read and write load on the SSD. N_{chip} is a constant for an SSD; N_{die} and P_a are constants for a NAND Flash. $N_{Tprogram}$ can be assessed as the number of $T_{program}$ on an R/B signal. M_p—the number of parallel planes in a die—can be obtained as the number of page programs in one die of one tested NAND Flash during a time period of $T_{program}$ in the SSD. Please refer to Section 3.1 for a way of measuring the number of page programs by the $T_{program}$ of an R/B signal.

3.4 Quantifying $N_{Tprogram}$

To obtain the value of $N_{Tprogram}$ in the model described in Section 3.3, we need to test a single R/B in one NAND Flash of an SSD (see the justifications in Section 3.2.3). The R/B signal levels are scanned when workload is loaded on the SSD. Without operations, the output level of R/B is high. When an operation begins, the level of R/B becomes low, the RB-Explorer system records the starting time T_s. When the R/B level changes to high, time T_e value is recorded (see Fig. 3). The low level's duration (i.e., $T_{program}$) is calculated by the system as $T_{program} = T_s - T_e$. If $T_{program} \in (a, b)\mu s$, one page is programmed. In doing so, the total number $N_{Tprogram}$ of $T_{program}$ on an R/B signal under a workload condition can be counted. Thus, the value of $N_{Tprogram}$ is obtained by our RB-Explorer.

3.5 Quantifying M_p

From (4), the value of M_p depends on the page program models (e.g., single-plane or multi-plane page program model). Our RB-Explorer follows a two-step process to determine the value of M_p from the measured write amplification.

Step 1: RB-Explorer executes a micro-benchmark with an incompressible sequential access and fixed amount (e.g., 2 GB) of written data on an SSD. RB-Explorer keeps track of the number $N_{Tprogram}$ of low levels for page programs in an R/B signal of one NAND Flash in the SSD. Applying our write amplification model (see (7) in Section 3.3), RB-Explorer obtains the value of write amplification (i.e., *WA*) for the SSD under this micro-benchmark using (8):

$$WA = \frac{N_{channel} \times N_{chip} \times N_{die} \times P_a \times}{L\_Volume\_Data} \times N_{T_{program}}. \tag{8}$$

Step 2: the value of M_p can be decided by the write amplification measured in Step 1. In particular, if the write amplification value is approximately 1, M_p is 1; otherwise, M_p is set to 2.

Now we explain the rationale behind Step 2. Given an SSD, our micro-benchmark featured with incompressible sequential accesses makes the write amplification close to one. In practice, whether the write amplification provided by (8) equals to one largely depends on $N_{Tprogram}$. Let us consider the following two cases:

Case 1: If the WA value computed by (8) is approximately 1, then the number $N_{Tprogram}$ of $T_{program}$ is equal to the number of page programs in a die. In this case, we conclude that only one page in a plane is programmed during one period of the R/B low level for page program. Thus, the NAND

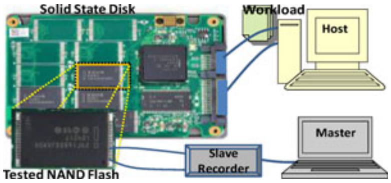


Fig. 4. The RB-Explorer system is to practically analyze the write amplification of an entire SSD rather than just NAND Flashes.

Flash is characterized as the one-plane page program model (i.e., M_p is 1, see also (4)).

Case 2: If the *WA* value obtained by (8) is around 0.5, then the number $N_{Tprogram}$ of $T_{program}$ is approximately half of the number of page programs in a die. This implies that two pages in two planes are concurrently programmed during the R/B signal's single period of low level for page program. Hence, the NAND Flash is expressed as the multi-plane page program model (i.e., M_p is 2, which is N_{plane} in (4)).

The above two cases allow RB-Explorer to determine which page program model (i.e., one-plane or multi-plane page program model) must be applied to NAND Flashes.

4 Validation of RB-Explorer

To validate RB-Explorer, we implement the RB-Explorer system to measure the $N_{Tprogram}$ value and the number of $T_{program}$ based on an R/B signal. We describe the measurement environment in Section 4.1. Section 4.2 outlines the approach to measuring $N_{Tprogram}$ in RB-Explorer. We also implement an SSD system to assess the accuracy and validity of our RB-Explorer (see Section 4.3). Furthermore and importantly, despite of the fact that SSD simulators have been the predominant tool used by researchers and developers alike to estimate write amplification, to the best of our knowledge, we argue that SSD simulators are far less flexible, practical, and accurate than a real SSD-based tool like RB-Explorer. To demonstrate this point, we compare our RB-Explorer to a fine-tuned SSD simulator, SSDsim, in terms of accuracy.

4.1 Measurement Environment

4.1.1 Experiment Platform

The RB-Explorer system (see Fig. 4) is composed of a host hardware platform, a master-slave recording system, a tested SSD, and two operating systems (i.e., Windows 7 and Ubuntu 11.10).

The host hardware platform is a desktop PC with an Intel Pentium 4 CPU with two cores, 2 GiB memory, and 1 TiB Western Digital disk, hosting the OSs. *The master* controls *the slave recorder* to keep track of the number of page programs in the selected NAND Flash based on logical data volume. When the workload and slave recorder come to an end of execution, values are transmitted to the master through a serial port. To verify the accuracy and credibility of our RB-Explorer, we implemented an in-house SSD called SSD-v, the features of which can be found in Table 2. The single-plane page program and die interleaved model is applied in all NAND Flashes of SSD-v. *Page-mapping*, *Cost-Benefit garbage collection*, and *static wear-leveling* are employed in the FTL of SSD-v. We also compare write amplification values measured by RB-Explorer with those obtained by a fine-tuned SSD simulator that also exploits the four-level full parallelism, SSDsim [17]. We configure SSDsim parameters to best resemble the SSD-v used in our experiments. Three real-world SSDs tested in this study include Intel X25-V (SSD-I) [10], Crucial m4 (SSD-M) [32], and SoliWare S80 (SSD-S). Using the RB-Explorer system as well as the NAND Flash datasheet, we obtain the interval value of $T_{program}$ (see Table 2).

TABLE 2
Tested SSDs and NAND Flash Characteristics

Product	**SSD-v**	**SSD-I**	**SSD-M**	**SSD-S**
Physical Capacity	16GB	40GB	64GB	32GB
User space	14.0GB	37.2GB	59.6GB	28.1GB
Over-provisioning	2GB	2.8GB	4.4GB	3.9GB
Cache Size	32KB	32MB	128MB	32KB
Flash Type	MLC 34nm		MLC25nm	SLC34nm
Program model	Single-plane		Multi-plane	
Chips	4	5	8	8
Chip Size	4GB	8GB	8GB	4GB
Channels (CHs)	4	5	8	4
Chips per CH	1	1	1	2
Dies per Chip	1	2	2	2
Planes per Die	2	2	2	2
Page Capacity	4KB+128Bytes		4KB+224bytes	
R/Bs per die	1		1	1
TRIM	Yes			
NAND Flash Typical Latency(Datasheet)				
Page read	20ns~50µs		75µs	25ns~25µs
Page program	900µs		1300µs	250µs
Block erase	2ms		3.8ms	2ms
NAND Flash Latency (Tested)				
Page program	(200~2200)µs		(200 ~2200)µs	(200 ~500)µs

4.1.2 Workload Characteristics

A wide range of micro-benchmarks and two real-world I/O traces [23] (see Table 3) are applied to test the write amplification of tested SSDs. The parameters in micro-benchmarks configured in the workload generator (i.e., *IOGenerator*) include read-write ratio, alignment of I/O on the SSD, the percentage of sequential or random accesses, runtime of workload, and data volume written from the host (i.e., logical written data). Note that the format of micro-benchmarks is expounded at the bottom of Table 3. *Blktrace* collects I/O traces, which are replayed by *IOReplayer*. The *IOGenerator* and *IOReplayer* are running on the host hardware platform.

4.2 Measuring $N_{Tprogram}$

The master-slave recording system of RB-Explorer selects a single NAND Flash in an SSD to test $N_{Tprogram}$ (see Fig. 4). The slave subsystem scans R/B signal levels when the benchmark is running on the SSD. Recall that (see Section 3.4) the total number $N_{Tprogram}$ of $T_{program}$ on an R/B signal under a given workload can be recorded by the slave recorder and transmitted to the master when the output of R/B remains at the high level for 5 minutes, which ensures that the page programs in NAND Flashes are fully completed. The pseudo-code for measuring $N_{Tprogram}$ by the master-slave recording system is described in the N_T_ PROGRAM_Procedure.

```
N_T_PROGRAM_ Procedure
Input:   $S_{R/B}$ = output level of R/B signal
$T_s$ = The beginning time of R/B signal being low
$T_e$ = The end time of R/B signal being low
$T_{program}$ = The duration of R/B signal for page program
SEND_OK = The time to send N_T_PROGRAM to the master
Output: N_T_PROGRAM    //the number of $T_{program}$
Step 1:  Initialize Ts, Te, SEND_OK, $T_{program}$ , and set i = 0
Step 2:  while (RB-Explorer is working) do
Step 3:          if (i==SEND_OK) then
                    send N_T_PROGRAM to the master
                    i = 0
                 fi
                 i++
Step 4:          if ($S_{R/B}$==1)//high level:1, low level:0
                    continue
                 fi
Step 5:          $T_s$=GetcurrentTime() //get the current time
Step 6:          while ($S_{R/B}$ ==0) do
                 od
Step 7:          $T_e$=GetcurrentTime()
                 i=0
Step 8:          if (($T_e$ - $T_s$)$\in T_{program}$)
                    N_T_PROGRAM = N_T_PROGRAM +1
                 fi
         od
```

4.3 RB-Explorer Verification

We compare true write amplification values (i.e., WA_{ture}) of SSD-v collected by its firmware with those measured by our RB-Explorer (i.e., $WA_{measured\_RB}$). We also obtain SSD-v's simulated write amplification (i.e., $WA_{measured\_SIM}$) offered by SSDsim. These three kinds of values are used to verify the accuracy of RB-Explorer under a workload condition. The relative error formula of write amplification (i.e., RE_{WA}) is given by

$$|RE_{WA}| = \frac{|WA_{measured} - WA_{true}|}{WA_{true}} \times 100\%. \tag{9}$$

The following two-step process is applied to obtain the true value of write amplification of SSD-v from its firmware and the measured one by RB-Explorer, respectively.

Step 1. We extend the firmware in SSD-v by incorporating a counter to keep track of the number of page programs executed by the SSD-v controller to NAND Flashes. The true number of page programs accurately offered by the counter in the SSD-v firmware is set as a parameter of Self-Monitoring, Analysis, and Reporting Technology (S. M.A.R.T). Using modified CrystalDiskInfo, an S.M.A.R.T. utility software, we obtain the true number of page programs in SSD-v.

Let N_{page_true} be the true number of page programs. Let P_a denote the page size of NAND Flash in SSD-v. Let *L_Volume_Data* represent the written data volume of the workload. The data volume written to NAND Flashes of SSD-v (i.e., *P_Volume_Data*) is a product of N_{page_true} and P_a. Applying N_{page_true} into (2), we obtain the true value of write amplification for SSD-v as

$$WA_{true} = \frac{P\_Volume\_Data}{L\_Volume\_Data} = \frac{P_a \times N_{page\_true}}{L\_Volume\_Data}. \tag{10}$$

TABLE 3
Workload Characteristics

Micro-benchmarks				
Write (%)	**Random (%)**		**Alignment**	**Avg. req. size**
100	0		4KiB-align	512Bytes, 4KiB, 8KiB, 16KiB, 32KiB
	100		4KiB-no-align	
50	0		4KiB-align	
	100		4KiB-no-align	

Real-word I/O Traces (Financial1 and Financial2)				
Traces	**Avg. req. size Write/Read (KiB)**	**Max. req. size Write/Read (KiB)**	**Written Data (GiB)**	**Write (%)**
F1	3.8/2.3	16K/8.3	14.6	77
F2	3.0/2.3	256/64	1.8	18

The format of Micro-benchmarks: [the percentage of Write I/Os in workload], [access type], [the size of alignment of each I/O in the tested SSD], [the size of I/Os]. Five tested I/O sizes are 512Byte, 4KiB, 8KiB, 16KiB, and 32KiB; two access types are 100% random (RD) and 0% random (i.e., 100% sequential , SQ); percentage of write IOs is set to 50% (50%-W) writes and 100% writes (100%-W); and alignment of each I/O on the disk is set to 4KiB-alignment (i.e., 4KiB-align) and 512B-alignment (i.e., 4KiB-no-align)). For example, a workload configuration of 4K-SQ-50%-W-4KiB-align means that the workload has the following characteristics: I/O request size is 4KiB, 50% writes are mixed with 50% reads, and 4KiB aligned blocks are sequentially issued to an SSD.

Step 2. The master-slave recording system in RB-Explorer (see Fig. 4) enables us to measure the number of page programs of all NAND Flashes in SSD-v. The pseudo-code presented in Section 4.2 is applied to count the number of page programs in our experiment. Let $N_{page\_measured\_RB}$ be the measured number of page programs in SSD-v by RB-Explorer. According to the characteristics of SSD-v in Table 2, it follows that $N_{chip} = 4, N_{die} = 1,$ *and* $M_p = 1$. Furthermore, from (2) and (7), we have

$$WA_{measured\_RB} = \frac{P\_Volume\_Data}{L\_Volume\_Data} = \frac{P_a \times N_{page\_measured\_RB}}{L\_Volume\_Data}, \tag{11}$$

where the value of $N_{page\_measured\_RB}$ is computed as

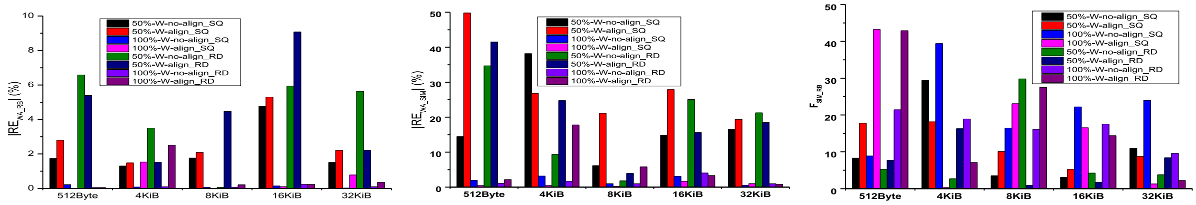
$$\begin{aligned} N_{page\_measured\_RB} &= N_{chip} \times N_{die} \times \left(N_{T_{program}} \times M_p\right) \\ &= 4 \times 1 \times \left(N_{T_{program}} \times 1\right). \end{aligned}$$

The $N_{Tprogram}$ can be measured by RB-Explorer's master-slave recording system. Combining (10) and (11), we rewrite (9) as

$$|RE_{WA\_RB}| = \frac{\left|N_{page\_measured\_RB} - N_{page\_true}\right|}{N_{page\_true}} \times 100\%. \tag{12}$$

Then, we compare N_{page_true} and $N_{page\_measured\_RB}$ under the same I/O load on SSD-v to verify RB-Explorer. Forty micro-benchmarks (see Table 3) are configured in *IOGenerator*. *L_Volume_Data* is set to 2 GB in the verification test; in other words, when the written data volume in workload reaches 2 GB, *IOGenerator* stops issuing I/O requests. In this process, the true number of page programs and the measured value of $N_{Tprogram}$ are recorded to compare N_{page_true} with $N_{page\_measured\_RB}$ using (12).

Fig. 5a reveals validation results of the relative error of write amplification, which demonstrates that our RB-Explorer is very accurate. For example, $|R_{EWA\_RB}|$ is smaller



(a)

(b)

(c)

Fig. 5. Relative error of write amplification: (a) $|RE_{WA\_RB}|$; (b) $|RE_{WA\_SIM}|$; (c) F_{SIM_RB} (i.e., $(|RE_{WA\_SIM}|)/(|RE_{WA\_RB}|)$).

than 1 percent (i.e., $|R_{EWA\_RB}| < 1\%$) in write-intensive micro-benchmarks, and $|RE_{WA\_RB}|$ is smaller than 10 percent (i.e., $|RE_{WA\_RB}| < 10\%$) in the read/write mixed micro-benchmarks. The validation results confirm the accuracy and credibility of our RB-Explorer. Thus, RB-Explorer can be employed to measure write amplification for SSDs under any I/O workload. The small value of $|RE_{WA\_RB}|$ is attributed to the following three reasons.

First, the typical latency erase is around 2 ms for NAND Flash in SSD-v. Interestingly, the latency for a program is anywhere between 200 and 2,200 μs, which makes program and erase time interleaved. Hence, the times of erases may be added to the number of programs. Thus, the error of an excessive number of programs may exit. Let us assume that the duration of page program distributes I.I.D. according to a uniform distribution as Unif [200, 2,200]. The probability of the duration of block erase being in the range from 200 to 2,200 is 10 percent; this probability is much smaller in real cases. In Fig. 5a, our empirical results confirm that the influence of interleaved duration of program and erase is insignificant.

Second, the sampling process for R/B signals may miss some low levels for page program, making the number of page programs smaller.

Third, there is a distinction between the program durations of lower-addressed pages and those of upper-addressed pages inside NAND Flash. Lower-addressed pages have shorter program durations than upper-addressed pages' program durations.

The plane-level parallelism, the lowest level of parallelism level, is independent of the other three levels of parallelism. The multi-plane model is physically distinct from the one-plane model. Nevertheless, the verification of the one-plane model can be used to verify the multi-plane model.

4.4 Improving on Accuracy of SSDsim

Now we demonstrate that RB-Explorer is more accurate than a state-of-the-art SSD simulator—SSDsim. We tune SSDsim's parameter to simulate SSD-v and collect SSD-v's simulated write amplification (i.e., $WA_{measured\_SIM}$). We run *IOGenerator* to drive both SSDsim and SSD-v; we assess the number of page programs (i.e., $N_{page\_measured\_SIM}$) from an output file for the comparison purpose. Comparing RE_{WA} *of* $WA_{measured\_SIM}$ and RE_{WA} *of* $WA_{measured\_RB}$, we conclude that RB-Explorer is more accurate than SSDsim in terms of measuring SSD write amplification.

Similar to (11) and (12), the relative error formula of write amplification for SSDsim is written as

$$|RE_{WA\_SIM}| = \frac{|N_{page\_measured\_SIM} - N_{page\_true}|}{N_{page\_true}} \times 100\%. \quad (13)$$

Fig. 5b plots the experimental results from the simulator. Except for the 4 KiB-100 percent-W-align_RD case, all the cases under the 100 percent-write workload have small $|RE_{WA\_SIM}|$ (i.e., $|RE_{WA\_SIM}| < 5\%$). Unfortunately, most $|RE_{WA\_SIM}|$ values in other cases are larger than 14 percent under the 50 percent-write workload. These values in Fig. 5b are much larger than those in Fig. 5a. In order to quantify the difference between $|RE_{WA\_RB}|$ and $|RE_{WA\_SIM}|$, we introduce

$$F_{SIM\_RB} = \frac{|RE_{WA\_SIM}|}{|RE_{WA\_RB}|}, \quad (14)$$

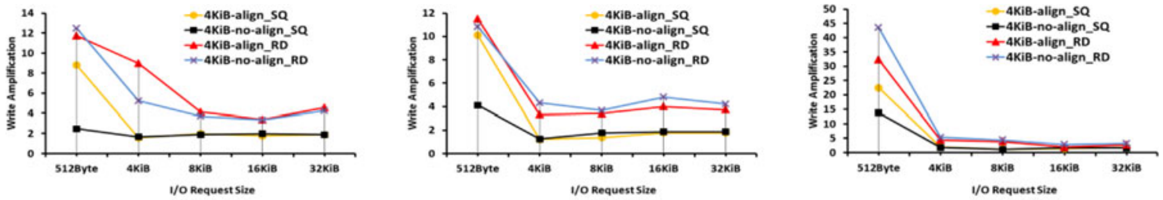
where F_{SIM_RB} is the deviation ratio of the write amplification value measured by the simulator to that measured by RB-Explorer. Obviously, the larger the F_{SIM_RB} values are, the more accurate RB-Explorer is than SSDsim. Fig. 5c reveals that in a few cases, SSDsim's accuracy is slightly higher than that of RB-Explorer, because SSDsim does not consider the interleaved time between program and erase operations, however small it may be. Fig. 5c also confirms that the accuracy of our RB-Explorer is much higher than that of SSDsim in most tested cases.

RB-Explorer improves on the accuracy of SSDsim because of the following three reasons: *First*, the event-driven-based SSDsim does not accurately resemble real-world SSDs from the perspective of bursty random-write I/Os; thus, SSDsim inaccurately simulates the frequency of garbage collection. *Second,* due to the initial mechanism for read operations in SSDsim, it issues a pre-write operation for each read request. Consequently, write data volume simulated by SSDsim is larger than that of the real-world SSD. *Third*, SSDsim is unable to simulate read disturb, which triggers excessive block erases and programs.

More importantly, since any one simulator cannot universally accurately simulate all SSDs, the fact that write amplification values measured by the fine-tuned SSDsim present a much larger deviation from true ones implies that simulators in general cannot flexibly and accurately measure the write amplification value.

5 Experimental Results

Now we employ RB-Explorer to evaluate write amplification and its performance impact of three real-world SSDs (i.e., SSD-I, SSD-M, and SSD-S) during the initial and steady states, respectively. Write amplification values



(a) SSD-I (b) SSD-M. (c) SSD-S.

Fig. 6. Fifty percent-write micro-benchmarks in the initial state of RAW SSDs.

measured in the initial state are reference points, and those obtained in the steady state represent standard ones for the tested SSDs.

We use Windows 7, on which *IOGenerator* is running, to partition the tested SSDs with 4 KiB alignment and to send a TRIM [15], [35] command to emulate SSD factory state (i.e., the initial state). *Blktrace* and *IORplayer*, running on Ubuntu 11.10, replay I/O traces. The slave recorder in RB-Explorer monitors an available R/B pin of one chip in tested SSDs to test its output level for page program. *IOGenerator* or *IOReplayer* stops issuing requests when the following two criteria are satisfied. *First*, the tested bandwidth measured in IOPS or MB/s reaches a stable value (see Section 5.1). *Second*, data volume written from the host satisfies a specified requirement (see Section 5.2 and Section 5.3). The slave recorder transmits $N_{Tprogram}$ to the master when the output of R/B remains at the high level for five minutes. The values of M_p are based on the program model of NAND Flash in tested SSDs in Table 2. From (7), we can obtain the write amplification values for the SSDs under the given workload conditions.

5.1 Micro-Benchmarks Measurement

Micro-benchmarks in Table 3 are all selected. We select two extreme workload conditions—full sequential and full random write cases, labeled “SQ” and “RD” respectively in the legends of the write amplification curves of Figs. 6, 7, 8, and 9. The I/O access patterns of other micro-benchmarks are between these two extreme cases. To evaluate the impact of read I/Os on write amplification, we choose 50 percent-read micro-benchmarks. Tested SSDs are opened with *O_DIRECT.*

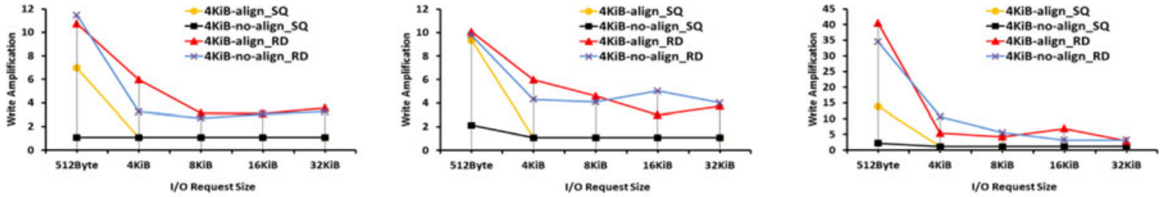
5.1.1 SSD Measurement in the Initial State

In the initial state, write amplification obtained by RB-Explorer is very small, because there is sufficient amount of available space for write I/Os to incur only infrequent garbage collection when data is updated. In additional, the values plotted in Figs. 6 and 7 are very low under sequential accesses. This result is attributed to the fact that SSDs take advantage of sequential accesses, which may incur less frequent but highly efficient garbage collection during page programs. Thus, these workload conditions help SSDs maintain low write amplification.

When the I/O size is small (see the 512-Byte results in Figs. 6 and 7), we found write amplification to be much higher than that under the sequential or random workload with big I/Os (e.g., I/O size from 4 to 32 KiB). Data volumes written per I/O transaction smaller than the page size (i.e., 4 KiB) trigger more page programs in NAND Flash, which increases write amplification. Our 512-Byte results also show that write amplification values under 4 KiB-align_SQ accesses are larger than those under 4 KiB-no-align_SQ accesses. The 4 KiB-align alignment is a burden in the case of sequential accesses of small I/Os. Because the alignment size of each I/O on the SSD is 4 KiB, they will begin at a multiple of 4 KiB from the beginning of an SSD. The size of the page is 4 KiB, but this type of I/Os cannot be programmed in the same page. In this case, the number of page programs increases and write amplification is larger under 4 KiB-align_SQ accesses than that under 4 KiB-no-align_SQ accesses.

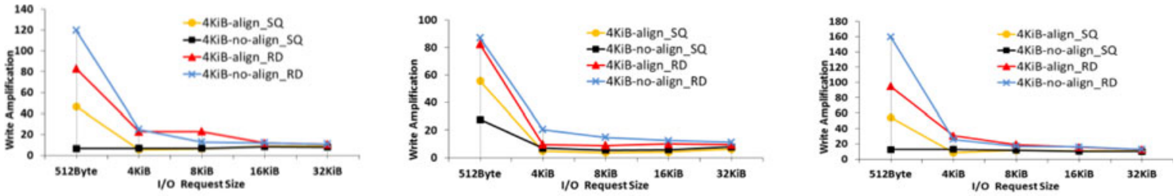
In Figs. 6a and 6b, write amplification under 512-Bytes I/O random accesses is smaller than that in Fig 6c. The cache space in SSD-I and SSD-M is 32 and 128 MB, respectively; and the cache is deployed to merge many small I/Os into larger ones to reduce the number of page updates and decrease write amplification accordingly. The cache space for SSD-S, 32 KB, is much smaller than that for SSD-I and SSD-M, resulting in much higher write amplification under 512-Byte random I/O accesses. However, the average values in the 8, 16, and 32 KiB benchmarks are much larger for SSD-M than SSD-I as shown in Figs. 6a and 6b; this may result from the different FTL technology.

When it comes to I/Os that are bigger than 512 Bytes (i.e., I/O size from 4 to 32 KiB), alignment hardly impacts the write amplification under sequential access patterns. The average value of write amplification ranges from 1.07 to 1.60 under the *4 KiB-no-align_SQ* accesses in Figs. 6 and 7. It



(a) SSD-I (b) SSD-M. (c) SSD-S.

Fig. 7. Hundred percent-write micro-benchmarks in the initial state of RAW SSDs.



(a) SSD-I (b) SSD-M. (c) SSD-S.

Fig. 8. Fifty percent-write micro-benchmarks in the steady state of RAW SSDs.

can be seen that garbage collection rarely occurs, giving rise to low write amplification.

Read I/Os noticeably affect write amplification when the cache space is small (see Figs. 6c versus 7c). For SSD-I and SSD-M, the impact of read I/Os on write amplification is less noticeable. Nevertheless, the write amplification values plotted in Fig. 6 are larger than those in Fig. 7. The reason for this trend is two-fold. *First*, a significant portion of DRAM is allocated to caching read data, limiting cache resources that may boost random write performance by merging small writes. *Second*, read operations in NAND Flash may cause the read disturb in blocks, meaning that the valid pages are rewritten before erasing the blocks. This enlarges write amplification.

5.1.2 SSD Measurement in the Steady State

To place the tested SSDs in the random-steady (steady for short) state, the 4 KiB-align_RD access is configured to distribute data across the available space of NAND Flashes as unevenly as possibly for about 12 hours.

In the steady state, Figs. 8 and 9 demonstrate that write amplification is very low in the case of sequential accesses. The average value of write amplification is anywhere between 1 and 5 except for the workload with 512-Bytes requests. In the process of random accesses, write amplification is between 10 and 50. The worst-case write amplification values all happen with 512-Byte I/O requests. In addition to small I/O sizes and random accesses, lack of sufficient data space for written data can aggravate write amplification. In this test, the capacity of over-provisioning (i.e., OP, the inclusion of extra storage capacity in NAND Flashes that is reserved only for SSD controllers) is an important factor affecting the write amplification for SSDs with less data space. OP lowers write amplification and makes the changes of values smooth (see the results from SSD-M).

Write amplification of SSD-M with a 128 MB cache varies slightly more than that of SSD-I with a 32 MB cache. SSD-S with a 32 KB cache becomes the worst case. The maximum write amplification changes from 85 to 90 for SSD-M (see Figs. 8b and 9b) but ranges from 110 to 150 for SSD-S (see Figs. 8c and 9c). This indicates that write amplification is sensitive to cache size.

Comparing with the 100 percent-write workload, the read I/Os in the 50 percent-write workload elevate write amplification in the steady state. Although write amplification decreases when I/O size increases, it is still higher than that under the 100 percent-write workload. Please refer to Section 5.1.1 for the reasons why read I/Os increase write amplification.

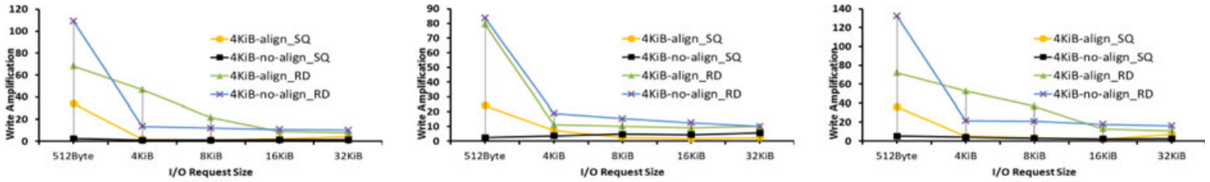
Since the tests for different SSDs are similar, our remaining tests are focused only on one SSD-I with $M_p = 1$.

5.2 Logical Data Volume in Partitioned SSDs

The logical data volume written from the host substantially affects write amplification. A large amount of data written to NAND Flash space leads to limited user-data space, which increases the probability of page updates and the frequency of garbage collection. This trend becomes pronounced under random workload conditions, which increases write amplification. Thus, the *relationship between write amplification of an SSD and logical data volume written from the host* (*WALVD*) is important in deepening this understanding and thus investigated in this paper.

In this test procedure, we focus on *WALVD* under five kinds of OP percent. Although the OP costs extra capacity in NAND Flashes, it is employed to reduce write amplification of SSDs. Under a given OP percent condition, we run the 4 KiB-align_RD access test for 15 times, in each of which the 5 GiB logical written data volume is configured by *IOGenerator*. After the amount of written data satisfies the requirement, the *IOGenerator* stops sending requests. The $N_{Tprogram}$ is transmitted to the master by the slave recorder when the output of R/B remains at the high level for five minutes. After the master receives the result, the next subtest will start. Fifteen sub-tests will be performed during each experiment. From (7), we derive write amplification from *L_Volume_Data* (5 GiB) and the value of $N_{Tprogram}$.

There is a *40* GB capacity in the partitioned SSD-I with *37.2* GB user space. Seven percent of *the total capacity* (i.e., 2.8 GB) is set as the *OP space*. The *available space*, in this



(a) SSD-I (b) SSD-M. (c) SSD-S.

Fig. 9. Hundred percent-write micro-benchmarks in the steady state of RAW SSDs.

TABLE 4
The Percentage of Over-Provisioning

OP%	*7%*	8%	10%	12%	15%
User_Space(GB)	*37.2 (default)*	32.2	25.2	20.5	15.9

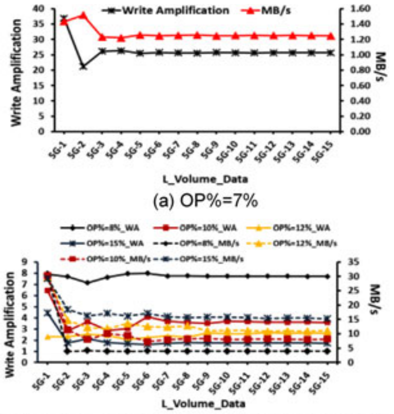
paper, includes *user space* and *OP space*. For SSD-I, the relationship between the OP percent in the available space of SSD and write amplification is given as

$$OP\% = \frac{OP \times 100\%}{Available\_Space} = \frac{OP \times 100\%}{OP + User\_Space} = \frac{2.8\,\text{GB}}{2.8\,\text{GB} + User\_Space} \times 100\%, \tag{15}$$

where OP space cannot be changed by users; however, the OP percent in the available space can be configured by modifying the user space size on the right-hand side of (15). Table 4 illustrates how we configure the user-space parameters. In each test, we initialize the SSDs to their factory default states by the TRIM command and partition according to the user-space size or *User_Space* (see (15)).

In Fig. 10 (*k* **in 5G-*k* standing for the *k*th sub-test**), write amplification at the point of *5G-1* is the highest among the entire test set, i.e., 39, 8, 6.5, 2.5 and 4.5 when OP percent is 7, 8, 10, 12, and 15 percent, respectively. These results are attributed to a process of initialization on the partitioned SSD-I, to which some initial data, not included in the *5* GiB data volume, must also be written. As long as the workload is loaded on SSD-I, the SSD controller will write the initial data to NAND Flashes, and the page program will occur. Then, the slave recorder begins recording the number $N_{Tprogram}$ of the R/B low levels for page programs in the tested NAND Flash. This data volume (i.e., initial data and 5 GiB logical data volume) is larger than that (i.e., 5 GiB) from the host in subsequent sub-tests; the $N_{Tprogram}$ is larger than that of the subsequent ones. Since the logical data volume is a constant, the value of write amplification is the largest in the initial state using (7).

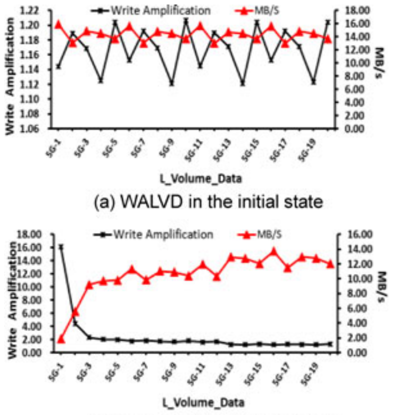
In Fig. 10a, the default value of OP percent requires independent manipulation. In Fig. 10b, we set the OP percent values according to Table 4. The write amplification drops to around 21, 3, 2.5, and 2 at the lowest point when OP percent is 7, 8, 10, 12, and 15 percent, respectively in the second (or third, e.g., OP% $= 8\%$) sub-test, in which only *5* GiB data is written from the host after the initialization phase. There is much more user space for random writes in an earlier sub-test than subsequent sub-tests. When the random data volume increases, free space decreases; OP hardly handles the increase of page-updates well. The write amplification increases with the logical data volume. The value of write amplification reaches a stable value (i.e., 25, 7, 3.5, 3, and 1 when OP percent is 7, 8, 10, 12, and 15 percent, respectively after the sixth to eighth sub-test), in which write performance varies inversely with write amplification. Fig. 10b also reveals that when the OP space in the available space of the partitioned SSD-I increases, the user space can take relatively more OP in the available space of the SSD; and thus, the possibility of garbage collection decreases under random workload. This improves performance by maintaining small write amplification.



(a) OP%=7%

(b) OP%=8%, OP%=10%, OP%=12%, OP%=15%

Fig. 10. WALVD under OP percent in partitioned SSD-I.



(a) WALVD in the initial state

(b) WALVD in the steady state

Fig. 11. WALVD under F1 in SSD-I.

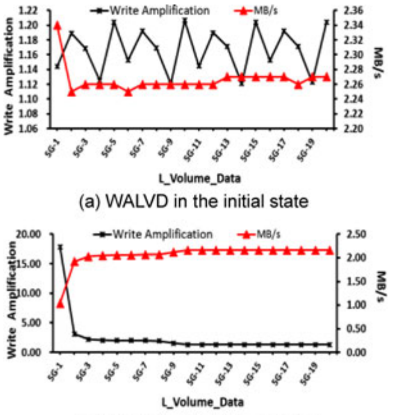
5.3 Real-World I/O Traces

We use the F1 and F2 traces to test write amplification and performance for SSD-I by RB-Explorer. Both F1 and F2 are real-world traces collected from OLTP applications [23]. These two traces contain many write requests; the average size of write requests is small (see Table 3). We study the impact of the read/write ratio on write amplification and performance under the trace condition.

We implement *IOReplayer* by extending the existing *Btreplay* in Ubuntu 11.10 for this group of experiments, in which data volume written from the host is configured. The *IOReplayer* replays the traces on the tested SSD-I, which is measured in both the initial and steady states (see Section 5.1 and 5.2 for details on the experimental setting). The *IOReplayer* stops issuing I/O requests after the amount of written data exceeds 5 GiB.

5.3.1 *The Financial1 (F1) Trace*

Fig. 11 shows the SSD-I write amplification and performance under the F1 trace. During the initial state (see Fig. 11a), write amplification varies from 1.12 to 1.22, depending on the increase of data volume, and write performance ranges from 13 to 16 MB/s. Write performance is inversely proportional to write amplification. Fig. 11b reveals write amplification and performance of the SSD in



(a) WALVD in the initial state

(b) WALVD in the steady state

Fig. 12. WALVD under F2 in SSD-I.

the steady state, in which write amplification is around 16 and write throughput is about 2 MB/s in the first subtest. The poor performance is attributed to less available space and a random data distribution.

In the subsequent sub-tests, the write amplification and performance become much better thanks to the write pattern in the trace. After the 13th sub-test, the write amplification and throughput become approximately 2 and 12 MB/s, respectively. When 77 percent of the requests are writes, the write performance is 16 and 12 MB/s in the initial and steady state, respectively. The write performance in the steady state is worse than that in the initial. The difference between write amplification in the initial and steady states is insignificant.

5.3.2 *The Financial2 (F2) Trace*

With 19 percent write-requests in F2, write performance is lower than those of F1 (see Fig. 12); however, the ideal range of write amplification is between 1.10 and 1.22 and the steady-state value is about 1.1. Like F1, the worst write amplification and performance in F2 are also experienced during the first sub-test in the steady state. The write amplification and throughput are approximately 17 and 1.0 MB/s, respectively. After about the 10th sub-test, the values reach a stable state.

These experiments confirm that sequential access patterns lead to low write amplification. In the initial state, the write amplification is low because of sufficient space for write I/Os. The write performance largely depends on access patterns (e.g., F1 has more write-I/Os than F2). In the steady state, the worst performance appears in the first subtest, which is a result of less space available for written data. A large number of read requests give rise to high write amplification and lower performance.

5.4 Evaluation Summary

Comparing the results reported in the prior studies with the experimental results measured by RB-Explorer, we draw the following conclusions:

- Our experiments show that the write amplification of an SSD (e.g., SSD-I) is smaller in the initial state (see Figs. 6 and 7) than that in the steady state (see Figs. 8 and 9). This observation is consistent to the results found in [42], [43], which stated that write amplification of an SSD was smaller in the initial state than that in the steady state due to more available user space and less garbage collection.
- The results published in [12] show that write amplification of an SSD is close to one under sequential I/O load and bigger than one under random I/O load, because the SSD takes full advantage of sequential accesses, which rarely trigger garbage collection. And this statement is confirmed by our experimental results. In Figs. 6, 7, 8, and 9, the write amplification values are smaller under the SQ workload than those under the RD workload; the values are close to one under the SQ workload when the I/O size is larger than 4 KiB.
- Hu and Haas [12] concluded that write amplification of an SSD is smaller under loads with large I/Os than that under loads with small I/Os. In our experiments (see Figs. 6, 7, 8, and 9), write amplification of the SSD processing 512-Byte I/Os is larger than that of the same SSD processing I/Os whose size is ranging from 4 to 32 KiB. Furthermore, the values under the 4 KiB-no-align-xx workload are smaller than those under the 4 KiB-align-xx workload in the 512-Byte experiments.
- We observe that write amplification of a tested SSD under the 50 percent-write workload is bigger than that under the 100 percent-write workload. This finding is consistent with the results found in [22], which reports that read-intensive I/O load gives rise to high write amplification of an SSD.
- The results plotted in Fig 10 show that the stable value of write amplification is 25, 7, 3.5, 3, and 1 when OP percent is 7, 8, 10, 12, and 15 percent, respectively. These results confirm those reported in [3] and [41], which state that the write amplification of an SSD with more OP space is smaller than the same SSD with less OP space.

The comparisons between our experimental results and the findings reported in the previous studies further validate the correctness of our RB-Explorer.

6 Related Work

Write amplification was initially proposed by Intel and Silicon Systems in 2008. Coulson [11] introduced a way of calculating write amplification. Hu et al. [3] suggested a probability analytical model to study the relationship between over-provisioning and write amplification in a simulator. A complex Markov chain model of SSD operations was developed by Bux [12] to explore the SSD performance using a page-level mapping scheme, which is inefficient to study write amplification. Wang [7] simplified the analytical model of write amplification for SSD with a page-level mapping mechanism. A closed-form expression for write amplification [4], [34] was mentioned by Agarwal and Xiang who improved the concept in a recent study [7], where a probabilistic model is presented to research the impact of over-provisioning on write amplification. Contrary to the above studies focused on probabilistic models, our work aims to develop a write amplification measurement method that can be validated by real-world SSDs. Our RB-Explorer pays attention to four parallelism levels, which were ignored in the existing models.

Various factors [12] affect write amplification of SSDs. A large DRAM capacity can merge small writes and decreases the frequency of out-of-place updates. Cache management schemes [2], [31], [38], [39], [40] are applied in SSDs to reduce random writes and page updates, thereby reducing write amplification.

Garbage collection (i.e., GC) reclaims free pages by erasing corresponding blocks. Wear-leveling mechanisms result in an even distribution of rewriting data across the NAND Flash. Both GC and wear-leveling give rise to large write amplification due to an increased number of page programs. A handful of mapping technologies (e.g., block, page, and the hybrid mappings) [1], [8], [9], [28], [29], [30], excellence garbage collection [14], [16], and wear-leveling [37] policies are optimized and enhanced to decrease excessive page programs. Over-provisioning or OP of NAND Flash capacity [15], being transparent to operating systems and applications, aims to improve SSD performance and endurance. The OP alleviates the overload of garbage collection to reduce write amplification. The TRIM [15], [35] command—a SATA command—is used to reclaim OP capacity to lower write amplification.

Data de-duplication [5], [13] and compression [18], [27] are also effective to eliminate data volume written to NAND Flashes. Read operations on NAND Flash may incur read disturb, which causes page rewriting, block erase, and data error. Hence, the ECC technologies [22], [40] were devised by He et al. to improve read disturb and data error. In addition, multi-level coding [36] reduces write amplification by rewriting page without the erase.

RB-Explorer proposed in this paper is targeted at offering a measurement toolkit for developers to evaluate the aforementioned techniques (e.g., GC, OP, TRIM, and ECC) that were developed to reduce write amplification.

7 Conclusions and Future Work

In this study, we proposed a new write amplification model and a novel approach, i.e., RB-Explorer, to measure write amplification in SSDs. We developed the SSD-v system and employed the fine-tuned-SSDsim to validate the credibility and accuracy of RB-Explorer. We also evaluated the measurement approach by performing a cross-comparison on a set of real-world SSDs. The validation results confirmed that our RB-Explorer was very accurate under a wide range of workload conditions.

By RE-Explorer, our findings show that SSDs exhibits low write amplification in the initial state, in which user space is sufficient and possibilities of garbage collection is very low. When random writes become a significant part of a workload condition, garbage collection frequently occurs, leading an increasing write amplification. The read operations also affects write amplification in two ways (i.e., limiting cache space for merging random small write I/Os; and causing read-disturb that triggers page-rewrites before erasing the blocks for data reliability) which enlarge write amplification. We observe that when it comes to partitioned SSDs during the initial state, initial data volume negatively affects write amplification; write amplification cannot noticeably change during the steady state. The over-provisioning in available space is also helpful in reducing write amplification; a large OP in available space offers low possibilities of garbage collection, which in turn reduce write amplification.

Our future work will concentrate on two areas: First, we will investigate the impacts of various components (e.g., data de-duplication, data compression, file systems, and I/O schedulers) not mentioned in this paper in a storage system on write amplification. Second, previous studies in the literature do not take black-box models into account while providing write amplification measurement. We plan to design a new black-box model and approach to simplify the testing procedure of SSD write amplification.

Acknowledgments

We are grateful to Xubin He, who works in Virginia Commonwealth University, for his constructive comments. We thank to the *SoliWare* Coporation providing assistance in designing the vSSD system. We also would like to thank to the anonymous reviewers for their suggestions on the earlier version of this manuscript. This work is sponsored in part by the National Basic Research Program of China under Grant No. 2011CB302303 and the National High Technology Research Program of China under Grant No. 2013AA013203, and the HUST Fund under Grant No. 2012QN101. This work is also supported by Key Laboratory of Information Storage System, Ministry of Education. Xiao Qin's work is supported by the U.S. National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0917137 (CSR), and CCF-0742187 (CPA). Hong Jiang's work is supported by US NSF under grants CNS-1116606, CNS-1016609, and IIS-0916859. Changsheng Xie is a corresponding author.

References

[1] S. Lee, D. Shin, Y. Kim, and J. Kim, "LAST: Locality-Aware Sector Translation for NAND Flash Memory-Based Storage Systems," *ACM SIGOPS Operating Systems Rev.*, vol. 42, no. 6, pp. 36-42, 2008.

[2] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber, "Extending SSD Lifetimes with Disk-Based Write Caches," *Proc. Eighth USENIX Conf. File and Storage Technologies*, Feb. 2010.

[3] X.-Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis, and R. Pletka, "Write Amplification Analysis in Flash-Based Solid State Drives," *Proc. SYSTOR'09: Israeli Experimental Systems Conf.*, May 2009.

[4] L. Xiang and B.M. Kurkoski, "An Improved Analytic Expression for Write Amplification in NAND Flash," *Proc. IEEE Int'l Conf. Computing, Networking and Comm. (ICNC)*, pp. 497-501, Jan./Feb. 2012.

[5] Q. Yang and J. Ren, "I-CASH: Intelligently Coupled Array of SSD and HDD," *Proc. IEEE 17th Int'l High Performance Computer Architecture (HPCA)*, pp. 278-289, June 2011.

[6] F. Masuoka, M. Momodomi, Y. Iwata, and R. Shirora, "New Ultra High Density EPROM and Flash EEPROM with NAND Structured Cell," *Proc. IEEE Int'l Electron Devices Meeting (IEDM)*, pp. 552-555, 1987.

[7] W. Wang, "A Simplified Model of Write Amplification for Solid State Drives Adopting Page Level Address Translation Mechanism," *Proc. Int'l Conf. Electrical Eng. and Automatic Control (ICEEAC)*, pp. 2156-2160, 2010.

[8] J. Kim, J.M. Kim, S.H. Noh, S.L. Min, and Y. Cho, "A Space-Efficient Flash Translation Layer for CompactFlash Systems," *IEEE Trans. Consumer Electronics*, vol. 48, no. 2, pp. 366-375, May 2002.

[9] A. Gupta, Y. Kim, and B. Urgaonkar, "DFTL: A Flash Translation Layer Employing Demand-Based Selective Caching of Page-Level Address Mappings," *Proc. ACM Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 229-240, Mar. 2009.

[10] Intel MD332 NAND Flash Memory Datasheet, June 2009.

[11] R. Coulson, "How Solid-State Drives Improve Computing Platforms,"*Intel IDF Fall*, 2008.

[12] X.-Y. Hu and R. Haas, "The Fundamental Limit of Flash Random Write Performance: Understanding, Analysis and Performance Modeling," research report, RZ 3771 (# 99781), IBM Research Zurich, Switzerland, Mar. 2010.

[13] J. Kim, C. Lee, S. Lee, I. Son, J. Choi, S. Yoon, H.-u. Lee, S. Kang, Y. Won, and J. Cha, "De-Duplication in SSDs: Model and Quantitative Analysis," *Proc. IEEE 28th Symp. Mass Storage Systems and Technologies (MSST '12)*, pp. 1-12, Apr. 2012.

[14] L.-P. Chang, T.-W. Kuo, and S.-W. Lo, "Real-Time Garbage Collection for Flash-Memory Storage Systems of Real-Time Embedded Systems," *ACM Trans. Embedded Computing Systems*, vol. 3, no. 4, pp. 837-863, Nov. 2004.

[15] T. Frankie, G. Hughes, and K. Kreutz-Delgado, "SSD Trim Commands Considerably Improve Overprovisioning," *Proc. Flash Memory Summit'11*, Aug. 2011.

[16] I. Iliadis, "Performance of the Greedy Garbage-Collection Scheme in Flash-Based Solid-State Drives," Research Report, RZ 3769 (# 99779), IBM Research, Zurich, Switzerland, Mar. 2010.

[17] Y. Hu, H. Jiang, D. Feng, L. Tian, H. Luo, and S.-P. Zhang, "Performance Impact and Interplay of SSD Parallelism through Advanced Commands, Allocation Strategy and Data Granularity," *Proc. Int'l Conf. Supercomputing (ICS '11)*, pp. 96-107, May 2011.

[18] T. Park and J.-S. Kim, "Compression Support for Flash Translation Layer," *Proc. Int'l Workshop Software Support for Portable Storage*, pp. 19-24, Oct. 2010.

[19] S. Park, E. Seo, J. Shin, S. Maeng, and J. Lee, "Exploiting Internal Parallelism of Flash-Based SSDs," *IEEE Computer Architecture Letters*, vol. 9, no. 1, pp. 9-12, Jan.-June 2010.

[20] F. Chen, R. Lee, and X. Zhang, "Essential Roles of Exploiting Internal Parallelism of Flash Memory Based Solid State Drives in High-Speed Data Processing," *Proc. IEEE 17th Int'l High Performance Computer Architecture (HPCA '11)*, pp. 266-277, Feb. 2011.

[21] Intel http://www.iometer.org/, 2014.

[22] S. Moon and A.L.N. Reddy, "Write Amplification Due to ECC on Flash Memory or Leave Those Bit Errors Alone," *Proc. IEEE 28th Symp. Mass Storage Systems and Technologies (MSST '12)*, pp. 1-6, Apr. 2012.

[23] K. B ates and B. McNutt, "SPC: Storage Process Council I/O traces," UMass Trace Repository, 2008.

[24] M. Jung, E. Herbert Wilson, and M.T. Kandemir, "Physically Addressed Queueing (PAQ): Improving Parallelism in Solid State Disks," *Proc. 39th Int'l Symp. Computer Architecture (ISCA '12)*, pp. 404-415, June 2012.

[25] White-box-testing, http://en.wikipedia.org/wiki/White-box_testing, 2014.

[26] A.M. Caulfield, L. M. Grupp, and S. Swanson, "Gordon: Using Flash Memory to Build Fast, Power-Efficient Clusters for Data-Intensive Applications," *Proc. ACM Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '09)*, pp. 217-228, Mar. 2009.

[27] G. Wu and X. He, "Delta-FTL: Improving SSD Lifetime via Exploiting Content Locality," *Proc. Seventh ACM European Conf. Computer Systems*, pp. 253-266, Apr. 2012.

[28] S.-W. Lee, D.-J. Park, T.-S Chung, D.-H. Lee, S.-W. Park, and H.-J. Song, "FAST: An FTL Scheme with Fully Associative Sector Translations," *Proc. US-Korea Conf. Science, Technology & Entrepreneurship*, Aug. 2005.

[29] J. Kang, H. Jo, J. Kim, and J. Lee, "A Superblock-Based Flash Translation Layer for NAND Flash Memory," *Proc. Sixth ACM/IEEE Int'l Conf. Embedded Software*, pp. 161-170, Oct. 2006.

[30] F. Chen, T. Luo, and X. Zhang, "CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory Based Solid State Drives," *Proc. Ninth USENIX Conf. File and Storage Technologies*, Feb. 2011.

[31] S. Kang, S. Park, H. Jung, H. Shim, and J. Cha, "Performance Trade-Offs in Using NVRAM Write Buffer for Flash Memory-Based Storage Devices," *IEEE Trans. Computers*, vol. 58, no. 6, pp. 744-758, June 2009.

[32] Micron MT29F64G08CFACB NAND Flash Memory Datasheet, Nov. 2011.

[33] Blktrace, http://linux.die.net/man/8/blktrace, 2014.

[34] R. Agarwal and Marrow, "A Closed-Form Expression for Write Amplification in NAND Flash," *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1846-1850, Dec. 2010.

[35] T. Frankie, G. Hughes, and K. Kreutz-Delgado, "A Mathematical Model of the Trim Command in NAND-Flash SSDs," *Proc. ACM 50th Ann. Southeast Regional Conf.*, pp. 59-64, 2012.

[36] A. Jagmohan, M. Franceschini, and L. Lastras, "Write Amplification Reduction in NAND Flash through Multi-Write Coding," *Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST '10)*, pp. 1-6, May 2010.

[37] D. Jung, Y.-H. Chae, H. Jo, J.-S. Kim, and J. Lee, "A Group-Based Wear-Leveling Algorithm for Large-Capacity Flash Memory Storage Systems," *Proc. Int'l Conf. Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, pp. 160-164, Sept. 2007.

[38] G. Wu, B. Eckart, and X. He, "BPAC: An Adaptive Write Buffer Management Scheme for Flash-Based Solid State Drives," *Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST '10)*, pp. 1-6, May 2010.

[39] H. Kim and S. Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage Abstract," *Proc. Sixth USENIX Conf. File and Storage Technologies*, May 2008.

[40] G. Wu, X. He, N. Xie, and T. Zhang, "DiffECC: Improving SSD Read Performance Using Differentiated Error Correction Coding Schemes," *Proc. Int'l Symp. Modeling, Analysis & Simulation of Computer and Telecomm. Systems (MASCOTS)*, pp. 57-66, Aug. 2010.

[41] K. Smith, "Understanding SSD over Provisioning," *Proc. Flash Memory Summit'12*, Aug. 2012.

[42] H. Mehling, "Solid State Drives Get Faster with TRIM," Enterprise Storage Forum, June 2010.

[43] T. Frankie, "Model and Analysis of Trim Commands in Solid State Drives," PhD dissertation, Dept. of Electrical Eng., UC San Diego, La Jolla, CA, 2012.

[44] http://en.wikipedia.org/wiki/Write_amplification, 2014.

Hui Sun is currently a PhD candidate in School of Computer Science and Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology (HUST), Wuhan, China. His research interest includes computer architecture, performance evaluation, NAND Flash memory-based storage systems, file systems, and I/O architectures.

Xiao Qin received the BS and MS degrees in Computer Science from the Huazhong University of Science and Technology, Wuhan, China, and the PhD degree in Computer Science from the University of Nebraska-Lincoln, USA, in 1992, 1999, and 2004, respectively. He is currently an Associate Professor with the Department of Computer Science and Software Engineering, Auburn University. His research interests include parallel and distributed systems, storage systems, fault tolerance, real-time systems, and performance evaluation. He was a recipient of the US National Science Foundation Computing Processes and Artifacts Award and the NSF Computer System Research Award in 2007 and the NSF CAREER Award in 2009. He is a senior member of the IEEE.

Hong Jiang received the BSc degree in Computer Engineering in 1982 from Huazhong University of Science and Technology, Wuhan, China; the MASc degree in Computer Engineering in 1987 from the University of Toronto, Toronto, Canada; and the PhD degree in Computer Science in 1991 from the Texas A&M University, College Station, Texas, USA. Since August 1991 he has been at the University of Nebraska-Lincoln, Lincoln, Nebraska, USA, where he is Willa Cather Professor of Computer Science and Engineering. At UNL, he has graduated 12 PhD students who upon their graduations either landed academic tenure-track positions in PhD-granting US institutions or were employed by major US IT corporations. His present research interests include computer architecture, computer storage systems and parallel I/O, high-performance computing, big data computing, cloud computing, performance evaluation. He serves as an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems. He has over 200 publications in major journals and international Conferences in these areas, including IEEE-TPDS, IEEE-TC, ACM-TACO, JPDC, ISCA, MICRO, USENIX ATC, FAST, LISA, ICDCS, IPDPS, MIDDLEWARE, OOPLAS, ECOOP, SC, ICS, HPDC, INFOCOM, ICPP, etc., and his research has been supported by NSF, DOD and the State of Nebraska. Dr. Jiang is a senior member of IEEE, and member of ACM.

Jianzhong Huang received the PhD degree in computer architecture in 2005 and completed the Post-Doctoral research in information engineering in 2007 from Huazhong University of Science and Technology (HUST), Wuhan, China. He is currently an associate professor in the Wuhan National Laboratory for Optoelectronics at HUST. His research interests include computer architecture and dependable storage systems. Dr. Huang was a recipient of the National Science Foundation of China in Storage System Research Award in 2007. He is a member of China Computer Federation (CCF).

Changsheng Xie received the BS and MS degrees in computer science both from Huazhong University of Science and Technology, Wuhan, China, in 1982 and 1988, respectively. He is currently a professor in the Department of Computer Engineering at HUST. He is also the director of the Data Storage Systems Laboratory of HUST and the deputy director of the Wuhan National Laboratory for Optoelectronics. His research interests include computer architecture, I/O system, and networked storage system. He is the vice chair of the expert committee of Storage Networking Industry Association (SNIA), China. Prof. Xie is a member of IEEE.