2013 IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems

Measuring and Analyzing Write Amplification Characteristics of Solid State Disks

Hui Sun[*], Xiao Qin[§], Fei Wu[*], Changsheng Xie[*†]
[*]School of Computer Science and Technology, National Laboratory for Optoelectronics
[*]Huazhong University of Science and Technology
[*]1037 Luoyu Road, Wuhan, China
[§]Department of Computer Science and Software Engineering
[§]Auburn University, [§]Auburn, AL 36849, USA
sunhuiworking@gmail.com, xqin@auburn.edu, cs_xie@hust.edu.cn

Abstract—*Write amplification* brings endurance challenges to NAND Flash-based solid state disks (SSDs) such as impacts upon their write endurance and lifetime. A large write amplification degrades program/erase cycles (P/Es) of NAND Flashes and reduces the endurance and performance of SSDs. The write amplification problem is mainly triggered by garbage collections, wear-leveling, metadata updates, and mapping table updates. Write amplification is defined as the ratio of data volume written by an SSD controller to data volume written by a host. In this paper, we propose a four-level model of write amplification for SSDs. The four levels considered in our model include the channel level, chip level, die level, and plane level. In light of this model, we design a method of analyzing write amplification of SSDs to trace SSD endurance and performance by incorporating the Ready/Busy (R/B) signal of NAND Flash. Our practical approach aims to measure the value of write amplification for an entire SSD rather than NAND Flashes. To validate our measurement technique and model, we implement a *verified SSD* (*vSSD*) system and perform a cross-comparison on a set of SSDs, which are stressed by micro-benchmarks and I/O traces. A new method for SSDs is adopted in our measurements to study the R/B signals of NAND Flashes in an SSD. Experimental results show that our model is accurate and the measurement technique is generally applicable to any SSDs.

I. INTRODUCTION

Write amplification has strong impacts on the endurance and performance of solid state disks (SSDs). This paper reports a model of write amplification for SSDs. The model considers parallelisms at four levels (i.e., the channel, chip, die, and plane levels). We also propose an approach to practically analyzing the write amplification of an entire SSD rather than just NAND Flashes. Our model and measurement solution for write amplification can be applied to trace SSD endurance and performance. We implement a system called vSSD to validate the accuracy and credibility of our model and approach.

SSD endurance depends on the limited number of P/Es in NAND Flashes. Increasing the number of available P/Es can substantially improve SSD lifetime. Write amplification is the ratio of data volume written by an SSD controller to data volume written by a host. High write amplification meaning a large number of page programs reduces available P/Es and degrades SSD endurance which is of importance for users. Write amplification not only enables users to quantify SSD endurance under any workload, but also offers ample opportunities to understand end-to-end implications of optimization strategies to boost SSD endurance. For example, I/O schedulers, file systems, and applications may have side effects on write amplification for SSDs. To obtain accurate SSD endurance and the influence of high level techniques on SSD, it is important to accurately evaluate or estimate the write amplification of SSDs. The lack of practical ways to measure write amplification for SSDs motivates us to propose a novel measuring method at the SSD level rather than the NAND Flash level. Our approach makes it possible to trace SSD endurance and to direct people to study and design excellent techniques to improve the endurance of SSDs. This method also evaluates novel technologies intended to reduce write amplification.

Due to out-of-place and erase-before-write updates in NAND Flash, reducing write amplification is a grand challenge for the development of NAND Flash-based storage devices. An out-of-place update prohibits rewriting an updated page at the same place. 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 excess P/Es or amplify page programs in NAND Flash, which yield lower endurance, shorten the lifespan of media, and reduce device performance.

Write amplification for an entire SSD is based on data volume written from a host, which is determined by I/O workload and can be technically assessed, and the data volume written to NAND Flashes, which can be obtained by our method. In our method, a tested SSD enclosure must be opened, and the output level of the R/B pin in one NAND Flash is tested. The duration of the low level of R/B varies with the different operations (i.e., read, program, and erase) in NAND Flash. The number of the low level of R/B for page program operations is calculated to gain the number of page programs. Data volume written to NAND Flashes by SSD controllers can be measured. The value of write amplification can be quantified. This paper makes three enabling contributions:

- *A new write amplification measurement approach:* I/O benchmarking tools focus on the performance of an SSD. These parameters, MB/s and IOPS which are obtained by these tools, depend on I/O workloads. A test is completed after a benchmarking loop finishes. The page program operations are also executed in NAND Flashes when a set of random I/O operations are issued. Unlike the existing

[†]Corresponding Author: Changsheng Xie. This work is sponsored in part by the National Basic Research Program of China (973 Program) under Grant No.2011CB302303, the National Natural Science Foundation of China under Grant No.60933002, and National High Technology Research and Development Program of China (863 Program) under Grant No.2013AA013203. Xiao Qin's work was supported by the U.S. National Science Foundation under Grants CCF-0845257(CAREER), CNS-0917137(CSR), and CCF-0742187(CPA).

1526-7539/13 $26.00 © 2013 IEEE
DOI 10.1109/MASCOTS.2013.29

tools, our proposed measurement approach is focused on SSDs rather than I/O workloads to gain data volume written in NAND Flash. We propose a four-level model of write amplification for SSDs. The four levels considered in our model include the channel, chip, die, and plane levels. We incorporate Ready/Busy (R/B) signals of an NAND Flash in this model. We develop a tool to evaluate SSD endurance by measuring write amplification. To improve our measurement approach, we also apply a new method, in which a tested SSD enclosure should be opened to measure R/B signals in the NAND Flash to obtain write amplification.

- *The new vSSD system:* We develop a system called *vSSD* to verify the accuracy and credibility of the model of write amplification and the proposed measurement approach. Our verification results show that the |PEN| value (|*PEN*|: Percentage Error of N_{page} in Section IV) is smaller than 1% in 100% write micro-benchmarks, and the |PEN| is smaller than 10% in mixed-write micro-benchmarks. This *vSSD* shows that our model is accurate and that the measurement technique is generally applicable to any 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 tested SSDs (e.g., the relationship between write amplification and performance). We investigate a *relationship between write amplification and data volume written by a host* (or *WALVD* for short).

The remainder of this paper is organized as follows. Background and related work of write amplification are provided in Section II. Section III discusses the measurement methodology and model for write amplification. The *vSSD* system and the methodology validation are presented in Section IV. Section V describes write amplification measurement results. The summary of our study and future work can be found in Section VI.

II. Background and Related Work

A. NAND Flash and Solid State Disks

NAND Flash memory is classified into three groups, namely, SLC (Single-Level Cell) NAND Flash, MLC (Multi-Level Cell), and TLC (Triple-level cell). A NAND Flash [10] is comprised of one or more targets, each of which is organized into one or more dies. A die is the minimum unit that can independently execute commands and report statuses by the 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, namely, read, program (write), and erase. A page is a basic unit for the read and program as is a block for the erase operation. In a block, a random page program is prohibited, and only out-of-place and erase-before-write updates are allowed when any of the pages need to be updated.

Solid state disks are mainly comprised of NAND Flash memory, a DRAM cache, and an SSD controller. DRAM improves performance of small data operations and temporarily stores a mapping table. The SSD controller is the most important component and contains an intermediate software layer called flash translation layer (FTL). FTL mainly performs address mapping, converting commands between a host and NAND Flashes, wear-leveling, garbage collection (GC), and ECC. Mapping technologies[3, 9, 23] are divided into block-mapping, page-mapping, and hybrid-mapping. Because out-of-place and erase-before-write updates cause an increasing number of P/Es, which reduces the endurance of NAND Flash, new hardware architectures and block management policies in FTL are designed for SSDs to extend the endurance of NAND Flash and prolong SSDs' lifetime.

B. Write Amplification

Write amplification was initially proposed for the Intel and Silicon Systems in 2008. Coulson [11], an Intel senior Fellow, introduced a way of calculating write amplification. Hu [1] suggested a probability analytical model to study the relationship between over-provisioning and write amplification. Hu also designed an ideal greedy reclaiming policy by the block-level address translation mechanism in a simulator. A Markov chain model of SSD operations was developed by Bux [12] to explore the performance characteristics of a system using a page-level mapping scheme, which is complex and inefficient for the research of write amplification. Applying a probability model, Wang [7] simplified the analytical model of write amplification for SSD with a page-level address translation mechanism. The closed-form expression for write amplification [21] was mentioned by Agarwal and Xiang [4] who improved the concept in a recent study where a probabilistic model is presented to research the impact of over-provisioning on write amplification.

Factors affecting write amplification include DRAM, types of workload (random, sequential, read, write), available user space, over-provisioning (OP) [1], mapping algorithms, garbage collection (GC) [14, 16], wear-leveling [6], TRIM [19] (a special SATA command), and error correct code (ECC) [22, 25]. The large DRAM capacity can merge small writes and decrease the frequency of out-of-place updates. Cache management schemes [2, 15, 26] are applied in SSDs to reduce random writes and out-of-place updates, which reduce write amplification. The GC operation, triggered by the out-of-place update in NAND Flash, reclaims free pages by erasing corresponding blocks. The wear-leveling results in an even distribution of rewriting data across the NAND Flash area. Both factors cause more page programs and increase write amplification. Excellent garbage collection and wear-leveling policies are optimized to reduce excess page programs. The over-provisioning or OP, reserved for SSD controllers to improve performance and endurance, is not accessible by operating systems and applications. The OP slows down the overload of garbage collection to reduce write amplification. TRIM clears the OP occupied to lighten write amplification. Data de-duplication [5, 13] and data compression [18, 27] are also effective to eliminate data volume written to NAND Flashes. Many read operations in NAND Flash can trigger read disturb, which causes page rewrite, block erase, and data error. Hence, some ECC technologies [25] were devised to improve read disturb and data error. In addition, multi-level coding [8], which allows page rewriting without erasing NAND Flash, reduces write amplification. Many studies only provide theoretical analysis by probabilistic models, which lack consideration of all realistic possible impacts on write

amplification. The aforementioned problem is addressed by our device-level model of write amplification for SSDs.

C. Quantifying Write Amplification

The definition of write amplification in this study is based on an entire SSD. Write amplification is calculated as the ratio of data volume written to the NAND Flashes by an SSD controller (physical) to data volume written from a host (logical). In Fig. 1, there are two kinds of data volume in the data stream from applications to NAND Flashes. The data volume written from the host under certain workloads must be stored in the SSD; this is called logical data (*L_Volume_Data*). When the SSD controller writes the total logical data to the storage pool of the NAND Flash, the data volume written to the NAND Flashes is called physical data (*P_Volume_Data*). Therefore, write amplification is given as

$$WA = \frac{P\_Volume\_Data}{L\_Volume\_Data}. \quad (1)$$

When the volume of logical data is equal to that of the physical one, the value of write amplification is one. When the value is larger than one, the physical data volume written to NAND Flashes is more than the logical data. Two types of data volume are used to calculate write amplification. The logical data volume can be known by users, and the data volume written to the NAND Flashes by a controller can be measured. Section III describes this method.

III. Measuring Write Amplification

In this study, we open an SSD's enclosure to test the R/B signal of one NAND Flash. The duration of a low level of R/B varies with different operations (i.e., read, program, and erase) in the NAND Flash. In our approach, we scan the output level of R/B to calculate the number of low level of R/B for page programs. This process obtains the number of page programs under a workload condition. Because the page size in one type of NAND Flash is constant, data volume written by a controller to an NAND Flash can be obtained as a product of the number of page programs and the page size of the NAND Flash. Thanks to the parallelism nature of NAND Flashes placed inside an SSD, the data volume written by the controller to all NAND Flashes can be easily obtained. This is a new approach to measuring the write amplification of SSDs under any I/O workload. In what follows, we explain the details of the new and practical method.

A. R/B signal in NAND Flash

Fig. 1 shows the existing T-SSD (i.e., a typical and abstract SSD architecture). R/B signals, one in each die, indicate the status of dies in NAND Flash. A low-level R/B signal means that an operation (e.g., read, program, or erase) in the die is in progress. An R/B pin of NAND Flash is an open-drain, active-low output, which uses an external pull-up resistor to observe the completion of program, read, and erase operations. The signal is typically at a high level during no operations and switches to a low level when any one starts. The duration of maintaining low R/B signals is different for three kinds of operations in NAND Flashes; the different low-level durations of R/B signals represent distinct activities. 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.

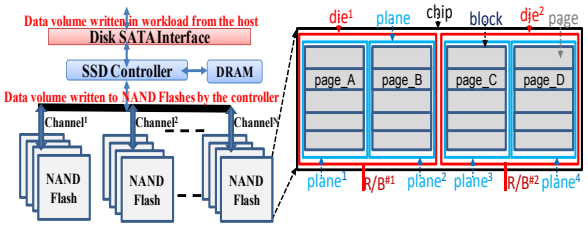


Fig. 1 One typical architecture of SSD (T-SSD)

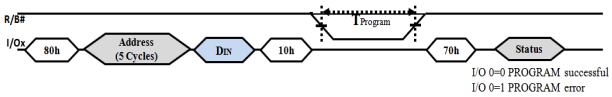


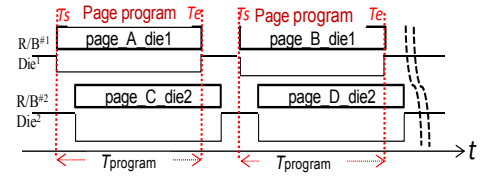
Fig.2 The timing diagram of a basic page program operation

Fig. 2 depicts the process of the basic page program under a single plane command in a plane of one die. First, it requires loading the 80h command (i.e., Serial Data Input) into the command register, followed by 5 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 in a plane of one die. 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. It means that one $T_{program}$ ($T_{program} \in (a, b)\mu s$) in one R/B signal contained in one die indicates one page program operation in a plane (see, for example, plane[1] of die[1] in Fig. 1). There is also a two-plane command, where one $T_{program}$ in one R/B signal contained in one die indicates two page programs in two planes (e.g. plane[1] and plane[2] of die[1] in Fig. 1). Addresses of the two pages programmed in two planes of one die must be identical. One R/B signal indicates operations in one die. According to the parallelism among dies, chips, and channels [17, 20, 24], the operation in one die can be implemented in other dies no matter which are in the same chip or not.

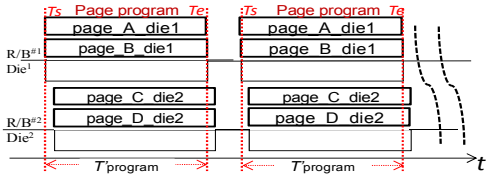
B. Parallelism and Program Models of SSDs

Parallelism improves SSD performance at four different levels, namely, channels-level, chips-level, dies-level, and planes-level (see also Fig. 1). The first three parallelism levels are applied to the SSD architecture. Sometimes, planes-level parallelism may not be applied in the single-plane model. There are two popular program models in NAND Flash. The first is the single-plane page program operation and die interleaved model; the second one is the multi-planes concurrent page program operation and die interleaved model.

The block diagram of a typical NAND Flash (two planes in one die, two dies in one chip) can be found in Fig. 1. In the first program model, 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 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 programming, 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



(a) Single-plane page program operation and die interleaved model



(b) Two-plane concurrent page program operation and die interleaved model

Fig.3 Two program operation models

parallelism (see Fig. 3(a)). The addresses of the two pages in two planes of one die are not restricted.

In the second model, $T_{program}$ of an R/B signal in a chip reflects two page programs 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 (see Fig. 3(b)).

According to the parallelism among chips (see Fig. 1), multiple chips on a channel run interleaved operations to guarantee that the interleaved operations among dies 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 channels-level parallelism makes the chips on multiple channels execute, in a parallel fashion, the same operation. Operations among chips on one channel are identical to ones on any of the channels. Regardless of the parallelism levels, it is accepted that an operation indicated from one R/B signal contained in one die is 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 one NAND Flash among all NAND Flashes within an SSD; the corresponding R/B signal is sufficient to conclude that all of the other NAND Flashes are active.

C. R/B Signal-Based Measurement

In (2), values of *L_Volume_Data* and *P_Volume_Data* must be given to derive write amplification. *L_Volume_Data* depends on I/O workload and *P_Volume_Data* is a product of the numbers of page programs and the page size. The page size is constant for a NAND Flash, and the number of page programs can be easily found. Using the parallelisms in an SSD system and the number of $T_{program}$ in one die for programming, we can determine the number of page programs for the entire SSD, in which the value of *P_Volume_Data* can be calculated. Next, write amplification of a tested SSD under certain I/O load can be measured.

$$WA = \frac{P\_Volume\_Data}{L\_Volume\_Data} = \frac{N_{channel} \times N_{chip} \times N_{die} \times N_p \times P_a}{L\_Volume\_Data}$$
$$= \frac{N_{channel} \times N_{chip} \times N_{die} \times (N_{T_{program}} \times M_p) \times P_a}{L\_Volume\_Data} \quad (2)$$
$$= \frac{N_{channel} \times N_{chip} \times N_{die} \times P_a}{L\_Volume\_Data} \times (N_{T_{program}} \times M_p)$$

$$M_p = \begin{cases} 1 & \text{single-plane page program operation and die interleaved model} \\ N_{plane} & \text{multi-planes concurrent page program operation and die interleaved model} \end{cases}$$

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 in T-SSD. One 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 in an interval between *a* and *b* μs. The total number of $T_{program}$, $N_{Tprogram}$, in one die under a workload condition should be recorded when page programs in NAND Flash are fully completed. Because there are two program models in NAND Flash (see Section III), the number of page programs in one $T_{program}$ is different in these two models. For *the single-plane page program operation and die interleaved model*, one page is programmed during each $T_{program}$ in a die. And two pages are programmed during each $T_{program}$ in a die based on *the multi-planes concurrent page program operation and die interleaved model*.

According to the dies-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 applies to any two dies of the same NAND Flash in T-SSD based on the chips-level and channels-level parallelism. The concurrency of $N_{channel}$ channels in T-SSD makes the relationship applicable to any two dies whether or not they are contained in the same NAND Flash. 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 *T-SSD* allows us to calculate the total number of page programs on the T-SSD under the workload. Under a workload condition, we need to count the number of $T_{program}$ ($N_{Tprogram}$) and obtain the number of page programs of each $T_{program}$, M_p, based on one R/B signal for one die in one tested NAND Flash. The total page programs during each $T_{program}$, N_p, can be calculated as a product of $N_{Tprogram}$ and M_p. Let P_a denote the size of one page; we can calculate the data volume written by a T-SSD controller to one die as $N_p \times P_a$. For the NAND Flashes in the T-SSD, there is one R/B pin per die. A chip, consisting of N_{die} dies, contains N_{die} R/B pins. And ($N_{chip} \times N_{die}$) R/B pins belong to the N_{chip} NAND Flashes per channel. Thus, there are ($N_{channel} \times N_{chip} \times N_{die}$) R/B pins in T-SSD with $N_{channel}$ channels. Given the parallelism of dies, chips, and channels, the relationship among N_{die} dies is identical, and the data volume written to all the NAND Flashes of the T-SSD is expressed as ($N_{channel} \times N_{chip} \times N_{die} \times N_p \times P_a$) or ($N_{channel} \times N_{chip} \times N_{die} \times (N_{Tprogram} \times M_p) \times P_a$). Using *L_Volume_Data*, P_a, $N_{Tprogram}$, and $T_{program}$, we can make use of (2), which is our write amplification model

TABLE I
CHARACTERISTICS TESTED SSDS AND NAND FLASHES

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
Overprovisioning	2GB	2.8GB	4.4GB	3.9GB
Cache Size	32KB	32MB	128MB	32KB
Flash Type	MLC 34nm		MLC25nm	SLC34nm
Program model	Single-plane		Two-planes-concurrent	
Chips	4	5	8	8
Chip Size	4GiB	8GiB	8GiB	4GiB
Channels (CH)	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	4KiB+128Bytes		4KiB+224bytes	
R/Bs per die	1		1	1
TRIM	Yes			
NAND Flash Typical Latency (Datasheet)				
Page read	20ns~50µs	75µs		25ns~25µs
Page write	900µs	1300µs		250µs
Block erase	2ms	3.8ms		2ms
NAND Flash Latency (Tested)				
Page write	(200~2200)µs	(200 ~2200)µs		(200 ~500)µs

based on four-level parallelisms (i.e., the channel level, chip level, die level, and plane level) of an SSD and the R/B signal in an NAND Flash to measure write amplification. *L_Volume_Data* in (2) can be measured, because data volume written from a host depends on reads and writes loaded on T-SSD. P_a is a constant for an NAND Flash; $N_{Tprogram}$ can be counted by the number of $T_{program}$ on an R/B signal. The number of page programs in one die of one NAND Flash tested in T-SSD can be assessed according to the $T_{program}$ of an R/B signal.

IV. EVALUATION METHODOLOGY

We design a measurement system to calculate the $N_{Tprogram}$ value and the number of $T_{program}$ based on an R/B signal. We describe the measurement approach in the context of two page program models in NAND Flashes. We also implement the vSSD (SSD-v) system to verify our proposed method.

A. Measurement Environment

The measurement system (see Fig. 4) is composed of a hardware platform, a master-slave recording system, vSSD, three tested SSDs, *IOGenerator* (a workload generator extended from IOmeter), *IOReplayer* (based on the Blktrace), and two operating systems (i.e., Windows 7 and Ubuntu 11.10).

The parameters configured in the workload generator include read-write ratio, alignment of I/O on SSD, the percentage of sequential or random accesses, runtime of a workload, and *data volume written from the host*. The platform is a desktop PC (i.e., hardware platform) with an Intel Pentium 4 CPU with 2 cores, 2GiB memory and 1TiB Western Digital disk, hosting the OSs. *Blktrace* collects I/O traces, which are replayed by the *IOReplayer*. The master controls the slave recorder to keep track of the number of page programs in NAND Flash selected based on logical data volume. When the workload and slave recorder come to a dead stop, values are transmitted to the master by a serial port. Table I summarizes the features of SSDs. The vSSD is designed to verify the accuracy and credibility of our device-level model of write amplification and measurement approach. Three real-world SSDs tested in this study include Intel X25-V SSD (SSD-I), Crucial™ m4 SSD (SSD-M), and SoliWare S80 SSD (SSD-S).

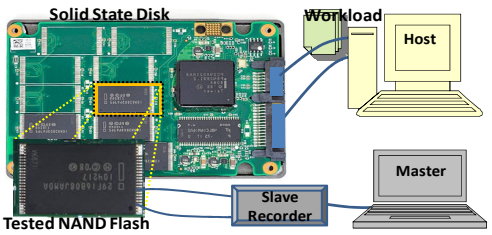


Fig. 4 The measurement system is to practically analyze the write amplification of an entire SSD rather than just NAND Flashes.

N_T_PROGRAM_Procedure

Input:
SR/B /*output level of R/B signal is the input parameter */
TS /*The time of R/B signal begin to be low*/
Te /*The time of R/B signal begin to be high*/
SEND_TIME /*The interval to send N_T_PROGRAM to master*/
SEND_TIME_OK /*Boolean, true, send N_T_PROGRAM to master */
$T_{program}$ /*The duration of R/B signal for program $T \in (a, b)\mu s$*/
Output:
N_T_PROGRAM /*the number of $T_{program}$ */
Begin
1: Slave_Recorder_Init(Ts, Te, SEND_TIME, T) /*initialize Ts, Te, T*/
2: while TRUE do
3: if (SEND_TIME_OK)
4: SEND_N (N_T_PROGRAM)
/*Send N_T_PROGRAM to the master by serial ports */
5: endif
6: COST_TIME(SEND_TIME) /*send-time elapses */
7: if (SR/B is high)
8: continue
9: end if
10: Ts=GetcurrentTime()
11: while SR/B =Low do /*waiting*/
12: end while
13: Te=GetcurrentTime()
14: RESET(SEND_TIME) /*reset the send-time point*/
15: if ((Te-Ts) ∈ T)
16: N_T_PROGRAM = N_T_PROGRAM +1
17: end if
18: end while
End

B. Measuring $N_{Tprogram}$

In the master-slave recording system, we only need to test a single NAND Flash in the SSD (see the justifications in Section III). The slave system scans R/B signal levels when I/O workload is loaded on the tested SSD. Without operations, the output level of R/B is high. When a page program begins, the level of R/B becomes low and the system records the starting time T_s. When the level of R/B changes to high, the time T_e is recorded (see Fig. 3(a)). The value of the duration of low level can be calculated by the system as $T_{program} = T_s - T_e$. If $T_{program} \in (a, b)\ \mu s$, one page program is executed. Afterward, the

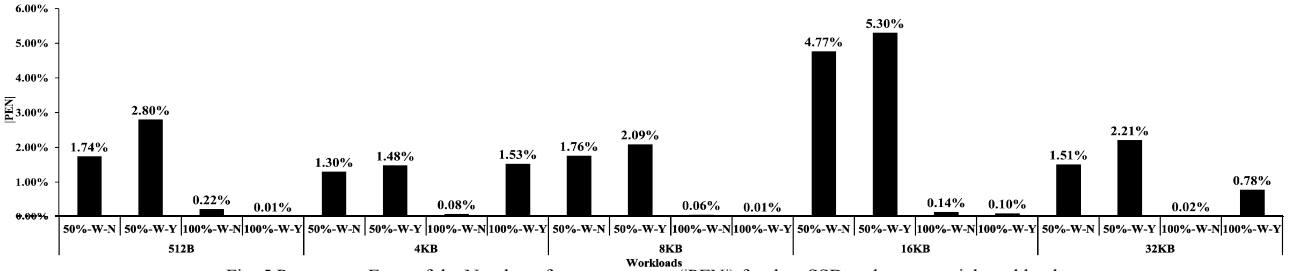


Fig. 5 Percentage Error of the Number of page programs (|PEN|) for the vSSD under sequential workloads

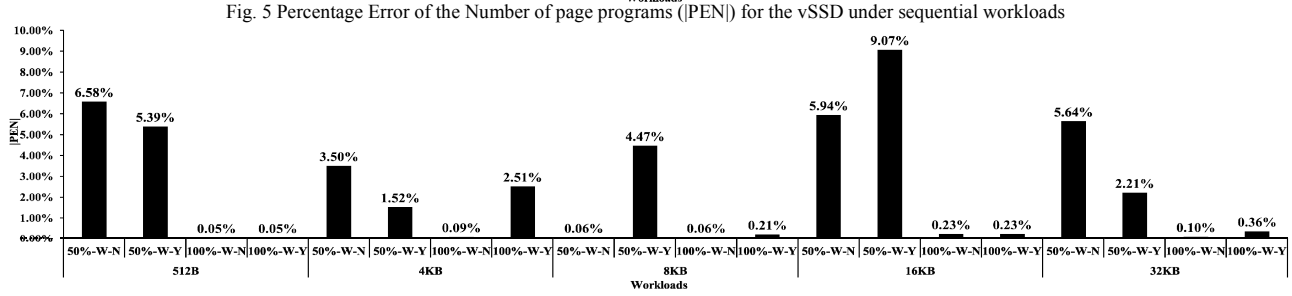


Fig. 6 Percentage Error of the Number of page programs (|PEN|) for the vSSD under random workloads

※*Workloads used in this verification test are configured in IOGenerator in the format described below: [I/O Size (512B, 4KiB, 8KiB, 16KiB, or 32KiB)]- [Access type (RD: random or SQ: sequential)]-[Percentage of write IOs in a workload (50%-W or 100%-W)]-[4KiB alignment of each I/O on the disk (Y: 4KiB-alignment or N: 512B-alignment)]*

master-slave recording system scans the R/B signal again. In doing so, the total number of $T_{program}$, $N_{Tprogram}$, in one die during a workload execution can be recorded by the slave recorder and transmitted to the master when the execution and slave recorder are fully stopped. The pseudo-code for measuring $N_{Tprogram}$ is displayed in the procedure of the *N_T_PROGRAM_Procedure*.

C. Method Verification

In cooperation with *SoliWare* (an SSD manufacturer), we designed a simple yet efficient vSSD to verify the accuracy of the proposed approach. The features of vSSD can be found in Table I. The single-plane page program operation and die interleaved model is applied in the vSSD. We modify the firmware in vSSD, adding and revising some special codes related to the page program, to recode the number of page programs by the vSSD controller. The value is set as a parameter of S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology). Using modified CrystalDiskInfo, an S.M.A.R.T. utility software, we obtain the number N_{page_real} of page programs for vSSD. With the master-slave recording system in place, we measure the number of page programs of all NAND Flashes, N_{page_test}, in the vSSD.

$$WA_{real} = \frac{P\_Volume\_Data}{L\_Volume\_Data} = \frac{P_a}{L\_Volume\_Data} \times N_{page\_real}$$

$$WA_{test} = \frac{P\_Volume\_Data}{L\_Volume\_Data} = \frac{N_{channel} \times N_{chip} \times N_{die} \times P_a}{L\_Volume\_Data} \times (N_{T_{program}} \times 1) \quad (3)$$

$$= \frac{P_a}{L\_Volume\_Data} \times N_{page\_test}$$

$$N_{page\_test} = N_{channel} \times N_{chip} \times N_{die} \times N_{T_{program}} = 4 \times 1 \times 1 \times N_{T_{program}}$$

$$|PEN| = \frac{\left| N_{\text{page\_test}} - N_{\text{page\_real}} \right|}{N_{\text{page\_real}}} \times 100\% \quad (4)$$

Write amplification can be expressed as (3). We compare WA_{real} with WA_{test} to verify the accuracy of our measurement system and the approach. (3) suggests comparing N_{page_real} and N_{page_test} rather than WA_{real} and WA_{test}. *Percentage Error of the Number of page programs (|PEN|)* is defined as (4). *L_Volume_Data* is set as 2GB in the verification test, in which the format of these micro-benchmarks is explained under Fig. 6. Figs. 5 and 6 present validation results, which demonstrate that our measurement system is very accurate. For example, |PEN| is smaller than 1% (i.e., $|PEN|<1\%$) in write-intensive micro-benchmarks, and |PEN| is smaller than 10% (i.e., $|PEN|<10\%$) in the read/write mixed micro-benchmarks.

Let us discuss why there are errors of the number of page programs in our approach. The typical latency of a program is around 900μs; the erase latency is about 2ms for NAND Flash in SSD-V (see Table I). Interestingly, the latency for a program is anywhere between 200 and 2200 μs in the tested NAND Flash**,** which makes program and erase time interleaved**.** Hence, the times of erase operations may be added to the number of programs. Thus, the error of an excess number of program operations may exit. Let us assume that the duration of page program distributes i.i.d. according to a uniform distribution as Unif [200, 2200]. The probability of the duration of block erase being in the range from 2000 to 2200 is 10%; this probability is much smaller in real cases. Our empirical results (see Figs. 5 and 6) confirm that the influence of interleaved duration of program and erase is insignificant. A second reason for the measurement errors is that the sampling process for R/B signals may miss some low levels for page program, making the number of page programs smaller. The third reason is the asynchronous program duration for lower pages and upper pages inside MLC NAND Flash. Nevertheless, the validation results confirm that the model of write amplification and measurement system is very accurate, indicating that the measurement system can be employed to measure write amplification under any I/O workload. The planes-level

TABLE II THE WORKLOAD CHARACTERISTICS

Percent Write IO in workload	Random/Sequential		Benchmark Alignment	I/O size (Bytes)
100% write	SQ (sequential)		4KiB-align	512, 4K, 8K, 16K, 32K
	RD (random)		4KiB-no-align	
50% write	SQ (sequential)		4KiB-align	512, 4K, 8K, 16K, 32K
	RD (random)		4KiB-no-align	

I/O Traces(Financial1 and Financial2)				
I/O traces	**Avg. req. size Write/Read (KiB)**	**Max. req. size Write/Read (KiB)**	**Written Data (GiB)**	**Read: Write(R:W)**
F1	3.8/2.3	16K/8.3	14.6	23:77
F2	3.0/2.3	256/64	1.8	82:18

Micro-benchmarks in this test follow the format explained under Fig. 6. The term, 4KiB-alignment (512B-alignment) of each I/O on the disk, is rewritten as 4KiB-align (4KiB-no-align). For example, "[100%write],[SQ],[4KiB -align],[8K]" represents that "[the percentage of write I/O in workload is 100% (no read I/O)],[the full sequential accesses],[4KiB alignment of each I/O in the SSD],[the size of I/O is 8KiB]".

parallelism, the lowest level, is independent of the other three levels. Multi-plane, a concurrent mechanism in a die, is physically distinct from one-plane. The verification of a one-plane model can be used to verify a multi-plane model.

V. PERFORMANCE ANALYSIS

We use Windows 7, on which *IOGenerator* (extended from IOmeter [29]) is executed, to partition the tested SSDs with 4KiB alignment and to send a TRIM command to emulate SSD factory state in the tests. Ubuntu 11.10, on which *Blktrace* [28] and *IORplayer* are running, is hosted on the same hardware platform to replay I/O traces. Only one R/B pin of one chip is selected to test the output level of its enclosing NAND Flash in the evaluated SSDs. According to (2), the number of $T_{program}$, $N_{Tprogram}$, must be recorded when the tested SSD reaches a stable stage. The execution time of the workload from beginning to stable state is so volatile that the workload sets in the *IOGenerator* continue running until the data volume written from a host satisfies the requirement. When program operations in all NAND Flashes stop, the value of $N_{Tprogram}$ in the tested NAND Flash is automatically recorded and transmitted to the master by the slave recording system. These values of M_p are different for tested SSDs.

A. Workloads

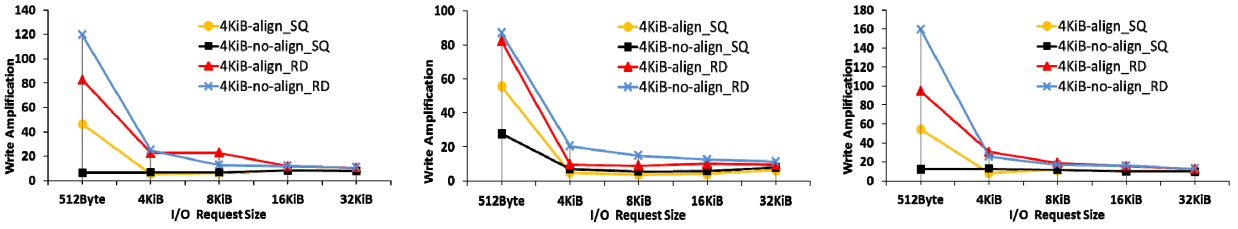
A wide range of micro-benchmarks and two real-world I/O traces (Financial1 and Financial2) are applied to evaluate the write amplification of tested SSDs. The characteristics of workloads are summarized in Table II.

B. Micro-benchmarks Measurement

Micro-benchmarks are configured in *IOGenerator* and I/O request size varied from 512 bytes to 32KB. We selected two extreme workload conditions—full sequential and full random write cases. SQ or RD in Table II stands for 100% sequential or 100% random workload, respectively. To evaluate the impact of read operations on write amplification, we choose micro-benchmarks with 50% writes and 50% reads.

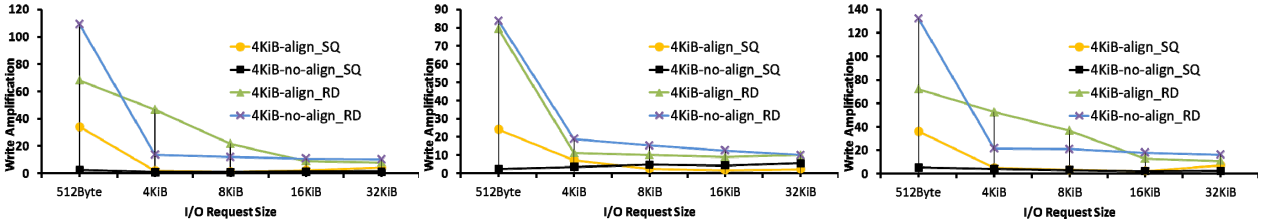
1) SSD measurement under steady state: A special micro-benchmark, 100% random writes with 4KiB-align I/Os, is configured to distribute data across the available space of NAND Flashes as unevenly as possibly for about 12 hours to place tested SSDs in the random-steady state. We run the workloads with 50% and 100% writes to measure the write amplification of the tested SSDs and to evaluate the impact of read operations (see Fig. 7). Figs. 7 and 8 confirm that write amplification is very low in the case of sequential writes. The average value of write amplification is anywhere between 1 and 5 except for the workload where request size is 512bytes. In the random case, write amplification is between 10 and 50. The worst case is the random workload with 512-byte I/Os.

Besides small I/O sizes and random workloads resulting in



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

Fig.7 50% write micro-benchmark in steady state of RAW SSD



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

Fig.8 100% write micro-benchmark in steady state of RAW SSD

high write amplification and poor performance, less data space for written data in this case can also cause detrimental results. In this test, the capacity of over-provisioning is important for SSDs. More over-provisioning and less garbage collection reduce write amplification. In the case of one tested SSD, more over-provisioning makes the changes of values smooth (see the results from SSD-M). The cache size also affects write amplification. For example, write amplification of SSD-M with a 128MB cache varies slightly more than that of SSD-I with a 32MB cache; SSD-S with a 32KB cache becomes the worst case in this test (see Figs. 7(c) and 8(c)). The write amplification results obtained in these sets of experiments represent the real or worse write amplification under sequential and random write micro-benchmarks.

Compared with the 100%-write workloads, 50%-write workloads increase write amplification. Although write amplification becomes small when I/O size decreases, the write amplification is larger than that of the sequential workloads. Compared to the small cache in SSD-I, the large cache space in SSD-M makes write amplification less sensitive to request size. For example, write amplification of SSD-M changes from 85 to 90; write amplification of SSD-I ranges from 110 to 150. Because of small data space, the over-provisioning (OP) size is an important factor to improve write amplification. The values of write amplification are smaller, and the trends of values are smoother for SSD-M with more OP than others. Read operations in the workload make write amplification worse. One reason for this trend is that a significant portion of DRAM is allocated to read, limiting cache resources that may boost write performance by merging small writes. Another reason can be attributed to the read disturb, which triggers some blocks to be updated frequently, thereby increasing page programs and making write amplification go up. The impact of read disturb becomes more pronounced for an SSD with less OP. The results of sequential workloads demonstrate that a large I/O size leads to small write amplification.

Regardless of the fact that the I/O size is in alignment to 4KiB, dramatic diversification in the values of write amplification rarely occurs except when there is a small I/O size (e.g., smaller than 4KiB) in the workload. Read operations in the workload make poor write amplification in the case of small I/O size. The results show that the tested values of write amplification are suitable for real-world workload conditions. Since the tests for different SSDs are similar, our remaining tests are focused only on SSD-I.

2) Logical Data Volume in Partitioned SSDs: The volume of logical data written to SSDs substantially affects write amplification. The user space in NAND Flashes of an SSD depends on the amount of data written by the controller. A large amount of written data to NAND Flashes leads to a small user space, which increases the probability of updating the same page and causes an increasing number of out-of-place updates and page programs. This trend becomes pronounced under random-write workload conditions, which increase write amplification. It is important to investigate *the relationship between write amplification and logical data volume written from the host (WALVD).*

Over-provisioning (OP) reserved only for SSD controllers is employed to reduce write amplification and improve performance and endurance of the tested SSD. There is a 40GB capacity in the partitioned SSD-I with 37.2GB user space. 2.8GB, 7% of the total capacity, is reserved by the manufacturer as the over-provisioning. Although OP costs storage capacity in NAND Flashes, it improves the write amplification and the performance of the SSD. The available space includes user space and OP space. In tested SSD, the OP space cannot be changed by users; however, the percentages of OP (OP %) in the available space can be configured. The relationship between the OP percentage in the available space of SSD is established as

$$OP\% = \frac{OP}{Available\_Space} = \frac{OP}{OP + User\_Space} = \frac{2.8GB}{2.8GB + User\_Space} \quad (5)$$

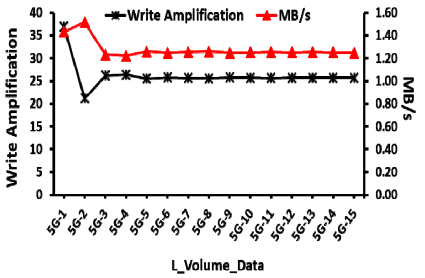
where we keep the OP space as a constant while changes the OP% value by modifying the user-space size on the right-hand side of (5). Table III illustrates how we configure the user-space parameters. In each test, we initialize the SSD to its factory default state by TRIM and partition according to the user-space size or User_Space in (5).

In this test, we focus on *the WALVD for five OP% values.* We run each test 15 times, in each of which the data volume written from the host is 5GiB. Note that this volume can be configured by IOGenerator. In Fig. 9, k in 5G-k stands for the k-th sub-test. Each workload is running until data volume written to the SSDs reaches 5GiB, at which point the *IOGenerator* stops issuing I/O requests. The number of page programs in the tested-NAND Flash is recorded by the slave recorder and transmitted to the master when the level of tested R/B signal keeps high for 5 minutes. After the master receives the result, the next sub-test (i.e., 5G-(k+1) test) will start. 15 sub-tests will be performed during each experiment. Using (2), we derive write amplification from *L_Volume_Data* (5GiB in sub-tests) and the number of page programs ($N_{Tprogram}$)

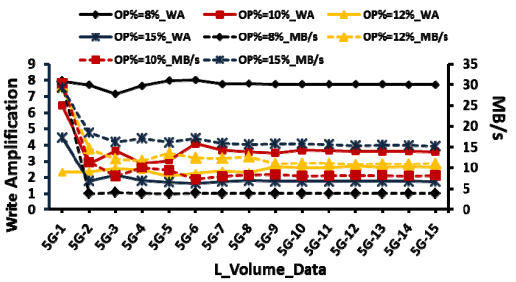
TABLE III

THE PERCENTAGE OF OVER-PROVISIONING

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



(a) WALVD under OP%=7%



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

Fig. 9 WALVD under different OP% in partitioned SSD

recorded by the slave recorder.

Fig. 9 plots experimental results for five OP% values. The results show that write amplification has the largest separation at the beginning of each test due to a process of initialization on the partitioned SSD, to which some initial data must be written. Prior to the first sub-test, the initial data had been loaded on the SSD; the initial data is not included in the logical data volume of 5GiB. As long as the workload is loaded on the SSD, the SSD controller will write the initial data to NAND Flashes and the page program will occur. When the initial data is loaded on the SSD, the slave recorder begins recording the number of page programs in the tested NAND Flash. Thus, the data volume, including the initial data and 5GiB logical data, is written to the SSD. This data volume is larger than that from the host in subsequent sub-tests; the number of page programs recorded is larger than that of the subsequent ones. Since the logical data volume is a constant (e.g., 5GiB), the value of write amplification is the largest in the initial state. For example, Fig. 9 shows that the write amplification at the point of *5G-1* is 39, 8, 6.5, 2.5 and 4.5 when OP% is 7%, 8%, 10%, 12%, and 15%, respectively.

In Fig. 9(a), OP%, set to the default value, requires independent manipulation. Write amplification measured in the first sub-test is the largest among the entire test; write amplification drops to 21 at the lowest point in the second sub-test, in which only 5GiB data is written from the host after the initialization phase. There is much more user space for the random writes in an earlier sub-test than subsequent sub-tests. When random data volume increases, free space decreases; the OP technique does not handle the decrease of free space well. Write amplification becomes large when the logical data volume increases. The value of write amplification reaches a stable value (i.e., about 25) after the 6th to 8th sub-test, during which write performance varies inversely with write amplification. For example, the write performance is 1.4 MB/s initially and becomes 1.5MB/s during the second sub-test. Write throughput is 1.23 MB/s during the steady stage. In Fig. 9(b), we set the OP% value according to Table III. A large value of OP% leads to a low stable value of write amplification. The average stable value is 7, 3.5, 3, and 1 when OP% is 8%, 10%, 12%, and 15%, respectively. Fig. 9 also shows that when the ratio of OP in the available space of partitioned SSD increases, the user space relatively owns more OP in the available space of NAND Flashes and reduces the possibility of out-of-place updates during random page programs. This reduces write amplification and improves performance.

C. *Real-World I/O Traces*

F1 and F2 are real-world I/O traces collected from OLTP (on-line transaction processing) applications currently running at two large financial institutions. These two traces contain many write requests, the average size of which is small (i.e., 3.8KB in F1 and 3.0KB in F2). F1 and F2 are used to test write amplification and performance. We study the impact of the read/write ratio on write amplification and write performance. We implement *IOReplayer* by extending the Btreplay in Ubuntu 11.10 for this group of experiments, in which data volume written from the host is configured. *IOReplayer* replays the OLTP I/O traces on the tested SSD, which are measured in the steady states. A micro-benchmark

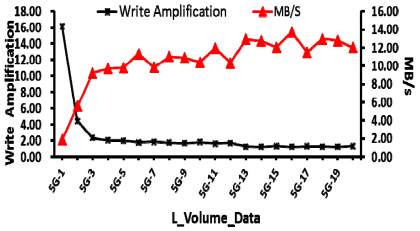


Fig. 10 WALVD in steady state for F1.

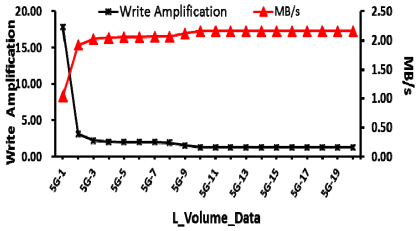


Fig. 11 WALVD in steady state for F2

with 100% random writes with 4KiB-align I/Os is issued to the tested SSD for a period of 12 hours and sets the SSD to the steady state.

1) F1 Trace: Fig. 10 shows the SSD write amplification and performance under the F1 trace. The SSD is in the steady state after running the trace of 4KiB-align write requests for 12 hours. Fig. 10 reveals the write amplification and write performance of the SSD in the steady state, in which write amplification is around 16 and write throughput is about 2MB/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 13 sub-tests, the write amplification and throughput become approximately 2 and 12MB/s, respectively. When 77% of the requests are writes, the write performance is 12MB/s in the steady state.

2) The F2 Trace: The experimental results under the F2 trace are similar to those under F1. With 19% of the requests being writes, write performance in F2 is lower than those of F1 (see Fig. 11); however, the steady 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 write throughput are about 17 and 1.0MB/s, respectively. After approximately ten sub-tests, the measured values reach a stable state. These experiments confirm that sequential access patterns (e.g., F1 and F2) lead to low write amplification. 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 sub-test, 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.

VI. Conclusions and Future Work

In this study, we proposed a new write amplification model and a novel approach to measuring write amplification in SSDs. We developed the *vSSD* system to validate the credibility and accuracy of our model. We also evaluated the approach by performing a cross-comparison on real-world

SSDs. These validation results confirmed that the measurement system is very accurate under a wide range of workload conditions. We made use of the measurement system to study the impacts of various micro-benchmarks and I/O traces on write amplification of SSDs. Our findings show that when random writes become a significant part of a workload condition, the out-of-place update frequently occurs, leading to an increasing number of page programs. A large number of page updates and block erases increase write amplification. The percentage of read operations also affects write amplification, which may cause read disturb that triggers rewriting of all pages in one block and block erasing for data reliability. DRAM can be used to merge small writes into larger ones to reduce write amplification. The over-provisioning in available space reduces write amplification; a large percentage of over-provisioning in available space offers low possibilities of garbage collection, which in turn improves write amplification and performance. We observed that write amplification and performance do not noticeably change during the steady state.

Our future work will concentrate on two areas. First, we will investigate the impacts of various components including file systems, I/O schedulers, SSD controllers, and parallelisms of SSDs in a storage system on write amplification. Second, we plan to design a new model and measurement system to simplify the testing of write amplification.

REFERENCES

[1] X.-Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis, and R. Pletka, "Write amplification analysis in flash-based solid state drives," *Proc. SYSTOR 2009: The Israeli Experimental Systems Conference*, May 2009.

[2] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber, "Extending SSD lifetimes with disk-based write caches," *Proc. eight USENIX Conf. File and Storage Technologies,* Feb. 2010.

[3] 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. nineth USENIX Conf. File and Storage Technologies,* Feb. 2011.

[4] L. Xiang, B. M. Kurkoski, "An Improved Analytic Expression for Write Amplification in NAND Flash," *Proc. IEEE International on Conference Computing, Networking and Communications (ICNC),* pp.497-501, Jan./Feb. 2012, doi: 10.1109/ICCNC.2012.6167472.

[5] Q. Yang and J. Ren, "I-CASH: Intelligently Coupled Array of SSD and HDD," *Pro. IEEE 17th International Symposium on High Performance Computer Architecture (HPCA),* pp. 278-289 Jun. 2011, doi:10.1109/HPCA.2011.5749736.

[6] 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. International conference on Compilers, architecture, and synthesis for embedded systems (CASES)*, pp. 160-164, Sep. 2007.

[7] Wei-Neng Wang, "A simplified Model of Write Amplification for Solid State Drives Adopting Page level Address Translation Mechanism," *ICEEAC,* pp.2156-2160, 2010.

[8] A. Jagmohan, M. Franceschini and L. Lastras, "Write amplification reduction in NAND Flash through multi-write coding," *Proc. IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp.1-6, May 2010, doi: 10.1109/MSST.2010.5496985.

[9] A. Gupta, Y. Kim and B. Urgaonkar, "DFTL: A flash translation layer employing demand-based selective caching of page-level address mappings," *Proc. the 14th International Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 229-240,* Mar. 2009.

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

[11] Rick 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] Kim, Jonghwa, et al. "De-duplication in SSDs: Model and quantitative analysis," *Proc. IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST),* pp. 1-12, Apr. 2012, doi: 10.1109/MSST.2012.6232379.

[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. Embed. Comput. Syst. (TECS)* Vol. 3, no.4, Nov. 2004.

[15] G. Wu, B. Eckart, X. He, "BPAC: An Adaptive Write Buffer Management Scheme for Flash-Based Solid State Drives," *Proc. IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp.1-6, May 2010, doi: 10.1109/MSST.2010.5496998.

[16] Ilias 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] Yang Hu , Hong Jiang , Dan Feng , Lei Tian , Hao Luo , and Shuping Zhang, "Performance impact and interplay of SSD parallelism through advanced commands, allocation strategy and data granularity," *Proc.* ACM *International conf. Supercomputing (ICS'11),* pp. 96-107, May, 2011.

[18] T. Park and J.-S. Kim, "Compression Support for Flash Translation Layer," *Proc. the International Workshop on Software Support for Portable Storage,* pp. 19-24, Oct. 2010.

[19] Tasha Frankie, Gordon F. Hughes, Kenneth Kreutz-Delgado, "A Mathematical Model of the Trim Command in NAND-Flash SSDs," *Proc.* ACM *50th Annual Southeast Regional Conference*, pp. 59-64, 2012.

[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 International Symposium on High Performance Computer Architecture (HPCA),* pp.266-277, Feb. 2011, doi: 10.1109/HPCA.2011.5749735.

[21] R. Agarwal, and Marrow, "A closed-form expression for Write Amplification in NAND Flash," In IEEE GLOBECOM Workshops (GC Wkshps), pp. 1846-1850, Dec. 2010, doi:10.1109/GLOCOMW.2010.5700261.

[22] S. Moon, A. L. Narasimha Reddy, "Write Amplification due to ECC on Flash Memory or Leave those Bit Errors Alone," *Proc. IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST),* pp.1-6, Apr. 2012, 10.1109/MSST.2012.6232375.

[23] J. Kang, H. Jo, J. Kim, and J. Lee. "A Superblock-based Flash Translation Layer for NAND Flash Memory," *Proc. the 6th ACM/IEEE International conference on Embedded software,* pp. 161-170, Oct. 2006.

[24] M. Jung, E. Herbert Wilson, M. T. Kandemir, "Physically Addressed Queueing (PAQ): Improving parallelism in Solid State Disks," *Proc. 39th International Symposium on Computer Architecture (ISCA)*, pp. 404-415, June 2012.

[25] G. Wu, X. He, N. Xie, T. Zhang, "DiffECC: Improving SSD Read Performance Using Differentiated Error Correction Coding Schemes," *Proc. IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS),* pp. 57-66, Aug. 2010, doi: 10.1109/MASCOTS.2010.15.

[26] 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.

[27] G. Wu and X. He, "Delta-FTL: improving SSD lifetime via exploiting content locality," *Proc. the 7th ACM European conference on Computer Systems,* pp. 253-266, Apr. 2012.

[28] Blktrace. http://linux.die.net/man/8/blktrace

[29] Intel, http://www.iometer.org/