Centaur: A Novel Architecture for Reliable, Low-Wear, High-Density 3D NAND Storage

CHUN-YI LIU, The Pennsylvania State University, USA
JAGADISH KOTRA, AMD Research, USA
MYOUNGSOO JUNG, KAIST, Republic of Korea
MAHMUT TAYLAN KANDEMIR, The Pennsylvania State University, USA

Due to the high density storage demand coming from applications from different domains, 3D NAND flash is becoming a promising candidate to replace 2D NAND flash as the dominant non-volatile memory. However, denser 3D NAND presents various performance and reliability issues, which can be addressed by the 3D NAND specific full-sequence program (FSP) operation. The FSP programs multiple pages simultaneously to mitigate the performance degradation caused by the long latency 3D NAND baseline program operations. However, the FSP-enabled 3D NAND-based SSDs introduce lifetime degradation due to the larger write granularities accessed by the FSP. To address the lifetime issue, in this paper, we propose and experimentally evaluate Centaur, a heterogeneous 2D/3D NAND heterogeneous SSD, as a solution. Centaur has three main components: a lifetime-aware inter-NAND request dispatcher, a lifetime-aware inter-NAND work stealer, and a data migration strategy from 2D NAND to 3D NAND. We used twelve SSD workloads to compare Centaur against a state-of-the-art 3D NAND-based SSD with the same capacity. Our experimental results indicate that the SSD lifetime and performance are improved by 3.7x and 1.11x, respectively, when using our 2D/3D heterogeneous SSD.

CCS Concepts: • **Hardware → External storage**; • **Software and its engineering → Secondary storage**.

Additional Key Words and Phrases: 3D NAND, Flash translation layer, Heterogeneous NAND

ACM Reference Format:
Chun-Yi Liu, Jagadish Kotra, Myoungsoo Jung, and Mahmut Taylan Kandemir. 2020. Centaur: A Novel Architecture for Reliable, Low-Wear, High-Density 3D NAND Storage. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 2, Article 28 (June 2020), 25 pages. https://doi.org/10.1145/3392146

1 INTRODUCTION

Solid state drives (SSDs) have become a popular storage option in many application domains, ranging from embedded computing, handheld devices to high-performance computing [16, 20, 27, 31, 40, 44, 58], largely because of their superior random access performance and compact form-factors. In fact, NAND-based SSDs can offer ultra high-density storage for space-constrained devices, such as compact computers, smartphones and tablets. For example, the maximum storage capacity of Apple iPhone [13] increased from 8 GB to 512 GB in 10 years, but its form factor did not improve significantly. This ever-increasing storage density is mainly achieved by reducing the 2D NAND memory node and increasing the number of stored bits per node. However, these

Authors' addresses: Chun-Yi Liu, cql5513@cse.psu.edu, The Pennsylvania State University, State College, PA, USA, 16801; Jagadish Kotra, Jagadish.Kotra@amd.com, AMD Research, Austin, TX, 78735, USA; Myoungsoo Jung, KAIST, Daejeon, 34141, Republic of Korea; Mahmut Taylan Kandemir, The Pennsylvania State University, State College, PA, 16801, USA.

2476-1249/2020/6-ART28 $15.00
https://doi.org/10.1145/3392146

density-increasing techniques have already reached the point of diminishing returns, as the higher bit density makes the 2D NAND flash less reliable, also shortening its lifetime. For example, the lifetime of TLC 2D NAND is $6x$ to $10x$ lower than that of multi-level cell (MLC) 2D NAND flash [26]. In addition, as the NAND technology shrunk from 50 nm to 20 nm, the bit error rate per NAND page increased by $416x$, making 2D NAND unreliable and requiring costly error corrections [49].

To address these challenges, the industry is embracing 3D NAND technology in various SSD systems. Instead of shrinking memory technology beyond the limit, 3D NAND *vertically* stacks charge trap or floating gate memory cells, thereby securing higher storage capacity with better reliability [32, 36, 37, 52, 56]. For example, 96-layer QLC-based 3D NAND [50] provides $21x$ higher density compared to 2D TLC NAND [47].

Although 3D NAND alleviates some of the reliability concerns in 2D NAND, it still possesses the 2D NAND multi-bit cell performance problems, and also introduces new reliability issues. Before discussing the issues in 3D NAND, it will help to explain the intricacies of a program operation in 2D and 3D NAND flash. Both 2D and 3D NAND flash enable only one bit to be programmed (written) at a time in a pre-defined order. The order is from bit 0 to bit $n-1$, where n stands for the number of bits per single cell (the n values for MLC and TLC are 2 and 3, respectively). The performance overheads in programming a multi-bit cell can be mainly attributed to two reasons: (a) redundant read operations of the bits programmed previously in the same cell, and (b) cell voltage computations that combine the old bits along with the new bit to program an additional bit. Hence, as the number of bits per cell increases further, the performance overhead involved in programming a multi-bit cell grows, thereby degrading the overall performance. On the other hand, the newly-introduced reliability issues in 3D NAND stem mainly from the page-indexing mechanism aggravated by the cell charge loss [24].

To overcome these 3D NAND issues, the 3D NAND manufacturers implement a technique called the *full-sequence program* (FSP) [32, 36, 37]. The FSP can program n bits, from bit 0 to bit $n-1$, into a cell "simultaneously", and its latency is nearly the same as the baseline program, since there is no necessity to read and compute the previous bits. Additionally, as the FSP stores data concurrently (averting the intermediate data), some of the reliability concerns [14, 18] in 3D NAND are automatically addressed by the FSP. However, utilizing the FSP in SSD brings up a *lifetime concern*. This is because the FSP operates in the order of 100 KiloByte granularity. As a result, it causes large amounts of data being written into the 3D NAND, irrespective of the original write request granularity, thereby effecting the SSD lifetime. One would think that this lifetime problem could be addressed by buffering the smaller write requests in an SSD-internal memory (DRAM) and accumulating the writes till they reach the FSP granularity. However, this mechanism is not a feasible solution as modern file systems [9, 11] and databases [8] force the storage systems to flush the DRAM-cached data to the storage media periodically (30 seconds or less) to prevent data loss due to power failures.

The FSP significantly increases the I/O unit size, making it larger than that in any 2D NAND technology. We observed that this large I/O unit size in turn wastes 91.6% storage space per page, and ultimately consumes $4.4x$ more lifetime than the conventional 2D NAND. While 2D single-level-cell (SLC) NAND does not have the multi-bit reliability issues that we have previously mentioned, the density of 2D SLC NAND is much smaller than that of 3D TLC NAND. Consequently, neither 2D nor 3D NAND alone can provide a reliable high-density compact SSD. To build a highly-reliable and high-density compact SSD, one promising option is, therefore, to *integrate* 2D SLC NAND and 3D TLC NAND.

In this paper, we propose *Centaur*, a novel heterogeneous architecture that integrates 2D SLC NAND and 3D TLC NAND within a "single" SSD device. Specifically, Centaur addresses the I/O unit size problem by guaranteeing that the small granularity write requests are processed by the

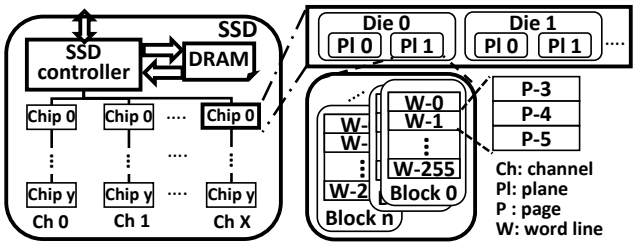


Fig. 1. Overview of a NAND-based SSD.

2D SLC NAND. Also, Centaur balances the lifetimes of the 2D NAND and 3D NAND via data migration and request redirection. Therefore, Centaur preserves the higher reliability and density of the 3D NAND, averting the lifetime problems caused by small granularity write requests. While Centaur is primarily optimized for lifetime, performance is also improved as a side-effect. *To our knowledge, this is the first work that integrates 2D NAND flash and 3D NAND flash in the same SSD.* Since lifetime optimization is the main target of Centaur and lifetime in SSDs is consumed by write operations, our main targeted workloads are *write-intensive* ones. Note however that read-dominated workloads are also slightly improved as a result of our approach.

The main **contributions** of this paper can be summarized as follows:

• It provides insight into how the performance and lifetime of 3D NAND are affected by the full-sequence program (FSP). Through our experimental evaluations, it is observed that the small granularity write requests have a severe negative impact on 3D NAND chips, in terms of both performance and lifetime.

• To address the lifetime issue caused by the FSP, we propose Centaur, a heterogeneous 2D/3D NAND SSD architecture. Centaur has three novel components: (a) a lifetime-aware inter-NAND request dispatcher, (b) a lifetime-aware inter-NAND work stealer, and (c) a data migration strategy from 2D NAND to 3D NAND. The dispatcher employs a dynamically-adjustable threshold to prevent small granularity write requests from being dispatched to 3D NAND. The inter-NAND work stealer balances the lifetimes through the request redirection from the 3D NAND chips to the 2D NAND chips, in case the 2D NAND or 3D NAND fails early. Finally, the data migration prevents the 2D NAND from running out of capacity, and it balances the lifetimes of the 2D NAND and 3D NAND by migrating the 2D NAND's data to the 3D NAND.

• We investigate the ideal chip layout for Centaur, and conclude that both NAND type chips should be evenly distributed across available channels to achieve better performance and longer lifetime. Our detailed experimental analysis indicates that, under iso-capacity, the lifetime and performance are improved by $3.7x$ and $1.11x$, respectively, over the baseline (state-of-the-art) 3D NAND flash, since the FSP-utilized 3D NAND chips in Centaur are *not* negatively impacted by small write requests. Further, our comparison of Centaur (which is a heterogeneous 2D/3D SSD) against a state-of-the-art SLC/MLC hybrid 2D SSD indicates 5x and 1.5x improvements, respectively, in lifetime and performance. To our knowledge, this paper presents the first detailed performance and lifetime evaluation of an integrated 2D/3D NAND flash.

2 BACKGROUND

2.1 SSD Organization

A NAND-based solid-state disk (SSD) [15, 33, 48] contains three basic components, namely, (a) a set of NAND chips, (b) a volatile memory, DRAM, and (c) an SSD controller, which are shown in Figure 1. The NAND chips are connected by channels, and each channel can transfer commands and data between the controller and a NAND chip independently to process the requests in parallel.

The NAND chips from different manufacturers share the same standardized interface and pin definitions, called Open NAND Flash Interface (ONFI) [3]. Hence, as long as the NAND chips are ONFI-compliant, they can be integrated into one SSD. Each NAND chip contains several dies, and each die has typically 2 to 4 NAND planes. A NAND plane is composed of n blocks, and n can be as large as 4,096, depending on the density of the NAND employed. Each block has multiple word lines, which are denoted by W-0 to W-m. A word line is composed of hundreds of thousands of NAND cells, which is up to $16K * 8$ cells per word line. In the triple-level cell (TLC) technology, one flash cell contains up to 3-bit information, and as a result, a 16KB NAND-cell word line provides three 16KB NAND pages, which are shown as P-3, P-4, and P-5 in Figure 1. A NAND cell can tolerate a limited number of program (write) and erase (clear) operations (PE cycle), ranging from 1,500 to 100,000. If the number of erase operations on a NAND cell exceeds this limit, the cell is worn out and can no longer store data. Current consumer TLC 3D NAND SSDs [4–6] can only be written 300 to 1,000 times [calculated by (Total Byte Written)/(SSD capacity)] in a 3-year warranty period. The number of times the new higher-density 3D NAND chips can be written is expected to be lower than the current ones.

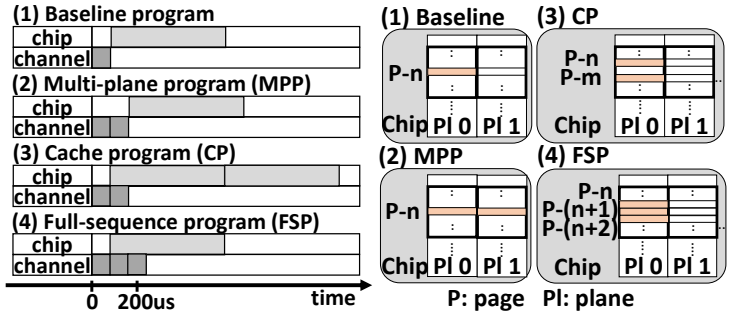
The volatile DRAM-based memory is employed in an SSD as a buffer to store the data being written into SSD and also other mapping meta-data viz, mapping table. The SSD controller, which runs the flash translation layer (FTL) [17, 28, 34, 39], mainly processes the read/write requests. The FTL provides the virtual-to-physical address translation demanded by the “erase-before-program" property of NAND flash. This constraint requires an erase operation to be performed before a page can be reprogrammed (rewritten). However, in SSDs, the erase operation is performed at a block-granularity. Hence, multiple pages have to be erased for the in-place page re-program (write). Therefore, an “out-place-update" strategy is used to reduce the erase operation overhead by writing the data into another page and tracking the location through an address translation mapping table. Modern SSDs mainly use page-level (mapping table) FTLs, since it provides the best performance and longest NAND chip lifetime. Though our design is based on page-level FTL, it can be extended to other FTLs with little modification.

Due to the "out-place-update" strategy adopted in SSDs, Garbage Collection (GC) algorithms are employed in the FTL to reclaim the invalid pages, which contain the older versions of data. Since the erase operation is at the block granularity, all valid pages in a to-be-erased block are copied to other blocks to prevent any loss of valid-data. More specifically, GC contains four steps: (a) selecting the victim block, (b) reading the valid pages from the victim block, (c) writing the valid pages to other blocks, and (d) erasing the victim block. In a page-level FTL, the simplest victim block selection algorithm is picking the block with the maximum number of invalid pages. More sophisticated selection algorithms can be found in [42, 55].

2.2 3D NAND Chip

2.2.1 The density

The high density of 3D NAND is achieved by two design decisions, (a) the layered architecture and (b) more bits per cell. As a result, a multi-layer 3D NAND accommodates more number of pages per block, and this increase in the number of pages per block in turn causes the “big block" problem [43, 57], where the number of valid pages to be copied during GC increases significantly, thereby increasing the GC overhead, and ultimately resulting in severe performance penalties. Besides the FTL management overheads, the layered architecture increases program disturbance [51] across the pages in different layers. On the other hand, increasing bits per cell allows more pages being programmed into one word line; however, to correctly program data, a more precise (but slower) cell programming technique needs to be implemented in 3D NAND. Both density-improved



(a) Latency. (b) Granularity.

Fig. 2. Overview of program operations.

designs increase the difficulty to correctly program data into cells; and as a result, the program latencies are significantly prolonged. Therefore, to mitigate such write performance degradation, the 3D NAND manufacturers implement the full-sequence program (FSP), which can program multiple pages simultaneously. Note that the FSP operation has been proposed long before the appearance of 3D NAND [21, 45], but the shorter voltage distribution of the 3D NAND cells has made the FSP implementation practical now.

2.2.2 Full-sequence program (FSP)

Before discussing the FSP, let us look at the various other program operations. Figures 2a and 2b show, respectively, the program latency and granularity of various program operations. The program latency can be broken into two parts: (a) data transfer time across the channel and (b) the program operation time inside the chip. Typically, the data transfer time is shorter than the program latency. The baseline program operation can only program one page in a plane, and the other planes are idle and cannot process any other operations meanwhile. To better utilize all the NAND die resources, a *multi-plane program (MPP) operation* is introduced, where the pages in the both planes can be programmed simultaneously with the constraint that the two pages have to be in the same relative positions in the corresponding planes or in the same block, depending on the manufacturer [1, 2]. *Cache program (CP) operation*, depicted in part 3 of Figure 2b, hides the data transfer time from the chip; so, a sequential page program operation to the same plane can have a slightly higher throughput, but the program latency for each page remains the same. The FSP operation is essentially an enhanced cache page operation, but the program latency is highly reduced due to the elimination of the redundant voltage checking across the page program operations. However, it has the constraint that the programmed pages must share the same word line. A detailed analysis of this is given later in Section 3.

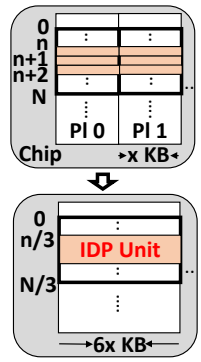
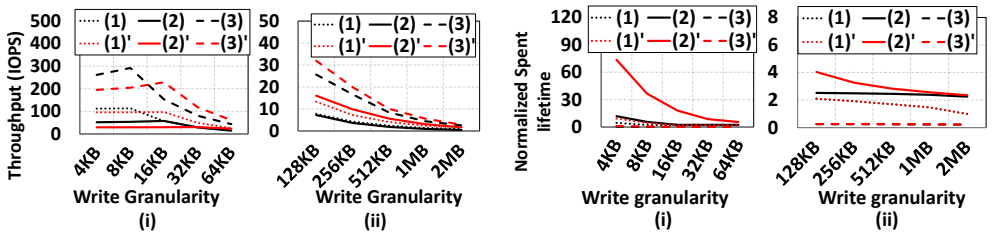


Fig. 3. The intra-die parallelism (IDP).

Different program operations can be combined together to provide better performance and higher reliability. For example, the MPP can be combined with the cache program and the FSP. Figure 3 illustrates an operation that combines the FSP and the MPP. This combined operation can fully utilize the intra-die parallelism (IDP) and achieve a very high throughput, but it also suffers from the same issues as the FSP. Hence, we refer to all the pages being programmed in the combined operation as "IDP (program) unit."



(a) Performance comparison. (b) Spent-lifetime comparison.

Fig. 5. Motivated chip configurations and comparisons.

3 MOTIVATION

As discussed in Section 2.2, utilizing the FSP in a 3D NAND can provide high performance [18], but the SSD lifetime may be negatively impacted. To provide an insight into how performance and lifetime are affected by the FSP and the distinct NAND dies, we performed single 2D single-level-cell (SLC) and 3D triple-level-cell (TLC) NAND die comparisons (iso-die count) with and without the *intra-die parallelism* (IDP), which is explained in Section 2.2.2. In addition, the performance and lifetime comparisons under the same total capacity (referred to as iso-capacity) are conducted as well. The reason to conduct iso-capacity comparisons is to show that a single 3D NAND die *cannot* outperform the same capacity multiple 2D NAND dies, owing to the degraded 3D NAND multi-die parallelism. That is, compared to a single 3D NAND die, multiple 2D NAND dies can process a higher number of read/write requests independently in parallel. However, a SSD with only a single 3D NAND die can be more compact than the one with multiple 2D NAND dies. With the iso-die count and iso-capacity comparison results, one can see the whole picture of the performance and lifetime impacts.

Figure 4 shows the various configurations being compared. (1) and (2) illustrate the block diagrams for the iso-die count configurations, while (2) and (3) give the iso-capacity configurations for the 2D and 3D NAND dies. The relevant parameters for the 2D and 3D NAND chips are given in Table 2. The performance and lifetime comparisons are conducted by using *random pattern write requests* with different granularities, from 4KB to 2MB. Figures 5a and 5b plot the throughput and lifetime comparisons, respectively, under all configurations with and without the IDP. The black lines indicate the configurations without the IDP, and the red lines correspond to those with the IDP. The 3D NAND IDP includes both the full-sequence program and multi-plane program operations; but, 2D NAND IDP has only multi-plane program operation due to the 2D SLC cell[1].

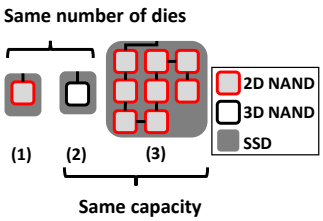


Fig. 4. Chip configuration.

Figure 5a-i gives the throughput[2] comparison with small write request granularities. Without the IDP, a 3D NAND die has a lower write throughput compared to that of a 2D NAND die, due to its longer write latency. Multiple 2D dies outperform others in throughput due to the multi-die parallelism. With the IDP, the throughputs of (1) and (3) increase when the write granularities are larger than the IDP unit, and the throughputs of these two configurations decrease when the write granularities are smaller than the IDP unit. This is because the write requests are amplified into larger granularities; as a result, fewer writes can be processed simultaneously. On the contrary, the

[1]We observed similar results with the 2D MLC cells as well. As a result, the comparisons between 2D MLC and 3D MLC are omitted.

[2]The IOPS is measured by the number of processed IOs per second. Hence, workloads with large writes, whose total write latency is longer, has a low IOPS compared to the workloads with small writes.

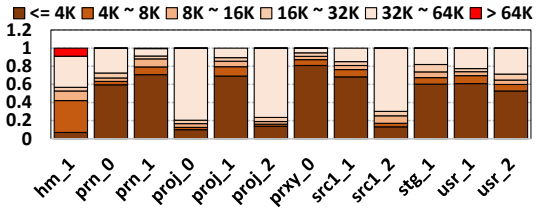


Fig. 6. The request granularity breakdown across workloads.

throughput of a single 3D die configuration drops to a minimal, owing to the much larger "IDP unit" granularity.

Figure 5a-ii shows the throughput comparison with the large write request granularities. Without the IDP, the trend is similar to the small granularity case. With the IDP, on the other hand, a single 3D NAND die outperforms a single 2D NAND die and shortens the performance gap between the multiple 2D NAND dies. In summary, the IDP improves the performance of large writes but degrades the performance of small writes.

Figure 5b plots the spent-lifetime[3] comparison under all the configurations considered, *with* and *without* the IDP. Let us define the spent-lifetime:

$$l_{2D} = \frac{\sum_{i=1}^{P_{2D}} \sum_{j=1}^{B_{2D}} e_{2Dij}}{P_{2D} * B_{2D} * E_{2D}}, \; l_{3D} = \frac{\sum_{i=1}^{P_{3D}} \sum_{j=1}^{B_{3D}} e_{3Dij}}{P_{3D} * B_{3D} * E_{3D}}. \tag{1}$$

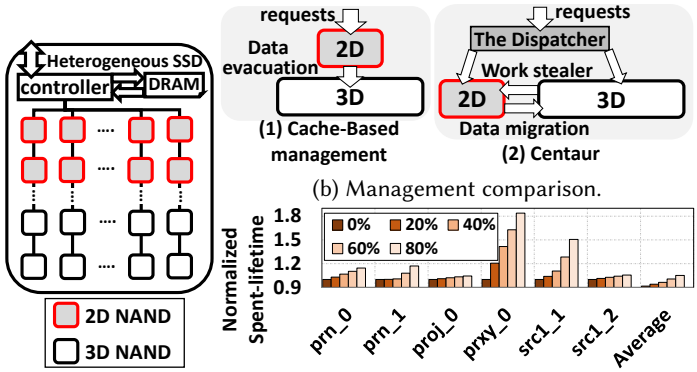
Equation (1) above is used to calculate the spent-lifetime of the 2D and 3D NAND chips, l_{2D} and l_{3D}, respectively. P_{2D} and P_{3D} are the total number of 2D and 3D planes, respectively. Note that one 3D NAND die contains only one plane instead two or more, since an "IDP unit" program operation always writes all planes in a die using the multi-plane operation. B_{2D} and B_{3D} in Equation (1) indicate the number of blocks per 2D plane and 3D plane, respectively, and E_{2D} and E_{3D} indicate the program/erase (PE) cycles that can be performed in a 2D and 3D block, respectively. Also, e_{2Dij} and e_{3Dij} represent the number of erase operations performed for the jth block in the ith plane, for 2D NAND and 3D NAND, respectively. Note that e_{2Dij} and e_{3Dij} are already maintained by modern SSDs to track whether a block is worn out.

The lifetime differences across the configurations are negligible except in the single 3D NAND die case with small write granularity, which is much worse than the other two cases. The reason is that the small granularity write requests are amplified to fit into the "IDP unit", so the program operations are performed with a small amount of valid data all the time, and consequently, the IDP-enabled 3D NAND die's lifetime is shortened dramatically compared to the one without the IDP. Therefore, *the small granularity write requests should avoid being executed by the IDP-enabled 3D NAND dies.*

To quantify whether the performance degradation and lifetime reduction brought by the small granularity write requests in the IDP-enabled 3D NAND can be important in practice, the fraction of small granularity write requests is plotted in Figure 6, where the real workloads from [38] are analyzed. More details about our workloads can be found in Table 1. The write request granularities of all the workloads are dominated by the request sizes smaller than 64KB, and some workloads, such as prn_1, prxy_0 and proj_1, are even dominated by the write granularities smaller than 4KB. Therefore, one can expect an IDP-enabled 3D NAND-based SSD to perform quite poorly under such workloads, in terms of *both* performance and lifetime.

Considering the results from Figures 4, 5a, and 5b, one can conclude that the IDP-enabled 3D NAND-based SSDs introduce both performance and lifetime problems, and the *lifetime problem is*

[3]The total written data across the workloads with different granularities are 24GB. We use "lifetime" and "spent-lifetime" interchangeably.



(a) Physical view. (c) Spent-lifetime affected by the amount of valid data.

Fig. 7. Overview of Centaur.

more severe than the performance problem. In addition, although the same-capacity 2D NAND-based SSD does not have the mentioned problems, the density of the 2D NAND-based SSD is much lower than the density of the 3D NAND-based SSD. Motivated by these observations, we next propose a *2D/3D heterogeneous SSD architecture* that accommodates *both* 2D and 3D NAND chips with the goal of achieving *both* longer lifetime and better performance. Note however that, as shown in Figures 5a and 5b, Centaur is mainly tuned to improve lifetime since, as stated above, the lifetime problem is more severe than the performance problem. The performance is improved as a side-effect of employing 2D SLC chips.

4 CENTAUR OVERVIEW

We propose Centaur to address the performance and lifetime issues caused by the 3D NAND chips with the enabled intra die parallelism (IDP), as discussed in Section 3. Since the lifetime issue is more severe than the performance issue, our design is mainly optimized for lifetime. A physical view of the proposed SSD architecture is provided in Figure 7a. Both 2D and 3D NAND chips are placed across all channels to fully utilize the multi-die parallelism; as a result, more requests can be serviced in parallel. We discussed and compared another possible NAND chip layout in Section 6.1. The 2D NAND used in Centaur is single-level-cell (SLC), while 3D NAND is multi-level-cell (MLC) or triple-level cell (TLC). The reason behind the decision is that only 2D SLC (not MLC or TLC) NAND performs better than 3D NAND in terms of *both* performance and lifetime. The main difference between 3D NAND in Centaur and conventional 2D NAND is that the IDP-enabled 3D NAND can use the FSP, which is not available in the 2D NAND.

In addition to the architectural changes, the FTL management in SSD controller in Centaur also needs to be modified, since the conventional SSD management is not designed for heterogeneous (2D and 3D) chips. Note that the naive cache-based management (shown in Figure 7b-1), which uses 2D as cache, is *not* a promising solution. This is because, when the limited-capacity 2D chips are full, there will be excessive data movements from 2D to 3D, and ultimately, the lifetime of 2D would significantly reduce. To be more specific, the larger requests, which can be processed by 3D chips without lifetime degradation, will fill out the 2D chips quickly; as a result, owing to the data evacuations, the requests will end up being redundantly processed by both 2D and 3D. Therefore, to prevent such lifetime degradation, Centaur should *dispatch some requests directly to 3D without passing through 2D.*

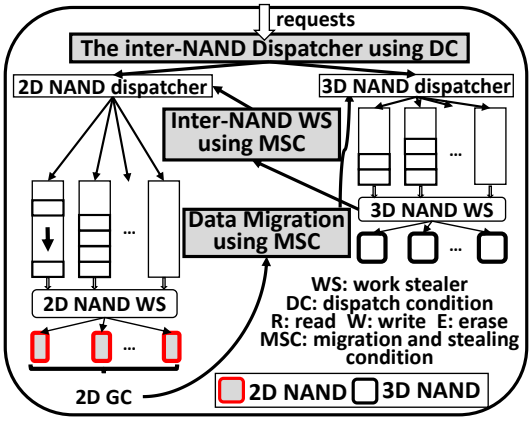


Fig. 8. The write request processing.

To address the mentioned lifetime issue, Centaur employs three different FTL functions, which are shown in Figure 7b-2: (a) a lifetime-aware inter-NAND request dispatcher, (b) a lifetime-aware inter-NAND work stealer, and (c) a data migration strategy. The dispatcher directs the requests to both 2D and 3D, according to a dynamically-adjustable *dispatch condition* (DC); hence, the excess data evacuations can be prevented. However, such a dynamic dispatcher may still not be optimal since it is agnostic to the number of queued requests per chip and the amount of valid data in the chips. As a result, it can introduce imbalanced loads across chips, which in turn shortens the overall lifetime. That is, a chip, which receives more writes, triggers more GC operations, and the GC frequency increases, as the amount of valid data in the chip increases, thereby degrading the lifetime. Figure 7c reveals that, under the same workload, a chip's lifetime is shortened significantly in the case of a larger amount of valid data. To address this issue, we propose a uni-direction *inter-NAND work stealer* and *data migration*. These two FTL functions are exclusively enabled by another dynamically varied threshold, referred to as *migration and stealing condition* (MSC). Therefore, a balanced amount of valid data can be achieved; hence, the lifetime balance between 2D and 3D can be improved, eventually prolonging the overall SSD lifetime. More details about the request indirection and the overheads are covered in Sections 5.3 and 5.4, respectively.

5 CENTAUR DETAILS

5.1 Lifetime-Aware Inter-NAND Dispatcher

Figure 8 depicts the request processing flow. Based on the dynamically-adjustable dispatch condition (DC), as depicted in Figure 8, our dispatcher sends the write requests to 2D or 3D (intra-)NAND dispatchers. The DC is determined by the *lifetime unevenness* (captured by Equation (1) in Section 3) between 2D NAND and 3D NAND. The goal of the dispatcher is to prolong the lifetime of 3D NAND by dispatching small granularity write requests to 2D NAND.

Note that the request granularity itself *cannot* serve as the DC since the write requests may not be aligned at the 3D "IDP unit" granularity; consequently, the small writes are still processed by "IDP unit." Therefore, our dispatcher uses the 3D "IDP unit"-aligned sub-request granularity as the DC instead. The example given in Figure 9 shows that the request granularity used as the DC can perform worse than the aligned sub-request granularity. In Figure 9, we assume that the DC is dynamically adjusted to 64KB. By using the request granularity as DC (Figure 9-1), the request A will be sent to 3D NAND, since 72KB is larger than the DC, 64KB. However, if we use the aligned "IDP unit" sub-request granularity as DC (as shown in Figure 9-2), the request A will be dispatched

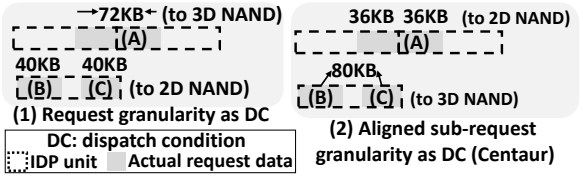


Fig. 9. The Dispatch condition (DC) example.

to 2D NAND, which can help us prevent the small granularity sub-request from being processed by 3D NAND, thereby improving the 3D NAND lifetime. On the other hand, in Figure 9-1, the requests B and C will be dispatched to 2D NAND separately. This is because 40KB is smaller than the DC. However, the requests B and C can be combined as one request, which has an 80KB data (Figure 9-2). Therefore, this combined request can be sent to the 3D NAND to balance the lifetimes of 2D and 3D NAND. Note that the requests are queued in the SSD waiting for processing; so, the data that belong to the same "IDP unit" are temporarily stored in the DRAM and can be combined to form a larger request.

Dispatch condition (DC): DC is a dynamically-adjustable "threshold" determined by the total lifetime spent by the 2D and 3D NAND. More specifically, a higher DC value means that the 3D NAND chips will receive fewer sub-requests and each sub-request will be nearly full of written data. Consequently, the 3D NAND chips' lifetime will be prolonged owing to a fewer number of program operations. However, the 2D NAND chips will receive more sub-requests, which shorten their lifetime. In contrast, a lower DC value will result in more small granular requests being dispatched to 3D NAND, and this will result in shorter 3D NAND lifetime. Hence, the DC plays a crucial role as far as the lifetimes of both 2D and 3D NAND are concerned. To that end, in Centaur, we adjust the DC value *dynamically* so that both 2D and 3D NAND chips will be worn out at the *same pace*. The definition of the spent-lifetime is shown in Equation (1) in Section 3. Note that Equation (1) only balances the coarser lifetime of 2D and 3D NAND, not the finer lifetime of the individual 2D/3D NAND chips. This is because the lifetimes of the homogeneous NAND chips are balanced by the intra-NAND work stealers [53, 54] and wear-levelers [23, 25, 41, 46]. More specifically, the intra-NAND work stealers are aware of the lifetime imbalance caused by the GC; so, the writes are redirected to the chips with fewer GCs.

Based on the lifetime discussion above, Algorithm 1 shows how the dispatcher adjusts the DC value. This algorithm assumes that the processed sub-requests are split from the requests and are aligned at a "IDP unit" granularity. *DspchCnt* stands for the number of requests dispatched since the last DC adjustment. The DC is only adjusted, once *DspchCnt* reaches *Adj_threshold*, where *Adj_threshold* is a preset threshold value. The lifetimes of the 2D and 3D NAND are calculated

Algorithm 1: The inter-NAND dispatcher

```
Input: sreq: the "IDP unit"-aligned sub-request
DspchCnt ← DspchCnt + 1;
if Adj_threshold ≤ DspchCnt then
    l_2D ← CALCULATE-LIFETIME();
    l_3D ← CALCULATE-LIFETIME();
    if l_2D < l_3D then
        if DC < DC_Max then DC ← DC + 512 Bytes;
    else
        if DC_Min < DC then DC ← DC − 512 Bytes;
    DspchCnt ← 0
if DC ≤ sreq.len then SEND-TO-3D-DISPATCHER(sreq);
else SEND-TO-2D-DISPATCHER(sreq);
```

Algorithm 2: INTER-NAND WORK STEALER

```
if v_3D > v_2D and IS-ANY-2D-CHIP-IDLE() == True then
    sreq_3D ← FIND-OLDEST-3D-WRITE-SREQS();
    if sreq_3D ≠ NULL then
        REMOVE-FROM-THE-3D-QUEUE(sreq_3D);
        sreqs_2D ←SPLIT-SREQ-TO-2D-GRANULARITY(sreq_3D);
        for sreq_2D in sreqs_2D do
            SEND-TO-2D-DISPATCHER(sreq_2D);
```

using Equation (1). DC_{Max} and DC_{Min} are preset values, where the following equation is satisfied.

$$0KB < DC_{Min} \leq DC \leq DC_{Max} \leq \text{“}IDP\ unit\text{”}\ size$$

5.2 The Lifetime-Aware Inter-NAND Work Stealer and Data Migration

5.2.1 The work stealer

The write sub-requests dispatched by the inter-NAND dispatcher using the DC are pushed further into the queues by the 2D/3D intra-NAND dispatcher [30, 35] (depicted in Figure 8), employed for the 2D and 3D chips. In Centaur, we employ an *inter*-NAND work stealer (shown in Figure 8) to balance the write sub-requests between the 2D and 3D NAND chips. Note that the *intra*-NAND (homogeneous) work stealers have already been proposed in [30, 53, 54] to improve the 2D and 3D NAND performances individually by balancing the number of queued write sub-requests. However, the homogeneous work stealers will *not* work well for the heterogeneous NAND design adopted in Centaur. This is because 2D and 3D NAND have distinct performance properties; as a result, 2D NAND can steal all requests from 3D NAND, which can in turn cause even severe lifetime problems.

Our inter-NAND work stealer considers the distinct properties of the NAND chips to prolong the lifetime of 3D NAND. The overall design of our inter-NAND work stealer has significant ramifications on the lifetime of both the 2D and 3D NAND chips. This is because, if too many write sub-requests are redirected from 3D NAND to 2D NAND, it will result in a large number of “out-of-place" updates in 2D NAND, causing in turn more GCs to be triggered, owing to the limited capacity of 2D NAND. This is why, in Centaur, we employ a *Migration and Stealing Condition* (MSC) which *dynamically* decides if the write sub-requests are to be redirected from 3D to 2D NAND, or the data already residing in 2D NAND need to be migrated to 3D NAND.

Our inter-NAND work stealer employs Equation (2) below to measure the percentage of valid data in both 2D and 3D NAND to take an informed decision regarding the request redirection and data migration. In Equation (2), v_{2D} and v_{3D} represent the average percentage of valid data across the 2D and 3D NAND, respectively.

$$v_{2D} = \frac{\text{2D valid pages}}{\text{2D total pages}}, v_{3D} = \frac{\text{3D valid pages}}{\text{3D total pages}}. \tag{2}$$

Based on the values of v_{2D} and v_{3D}, MSC enables either write request redirection *or* data migration. If $v_{2D} > v_{3D}$, data migration is triggered from 2D to 3D NAND. However, if $v_{3D} > v_{2D}$, the write sub-requests are re-directed from 3D to 2D NAND. Such data migration/write redirection by MSC reduces the number of GCs on 2D/3D NAND since it balances the percentages of valid data across 2D/3D NANDs.

Algorithm 2 gives the write redirection algorithm employed in Centaur when $v_{3D} > v_{2D}$. The granularity at which the write request redirection is triggered plays a crucial role in Centaur. Though it is possible to trigger write redirection whenever $v_{3D} > v_{2D}$, it may not always be

Algorithm 3: MIGRATION IN 2D NAND GC

```
   Input: vblk: the victim 2D NAND block
1  if v_2D > v_3D then                                              // Data migration
2      for page in vblk do
3          if IS-VALID-DATA(page) then
4              data ← DO-2D-READ-SREQ(page);
5              pages ← GET-OTHER-RELATED-2D-PAGES(page);
6              for p in pages do
7                  data_p ←DO-2D-READ-SREQ(p);
8                  data ←COALESCE(data, data_p)
9              DO-3D-WRITE-SREQ(data);
10 else                                                             // Baseline GC
11     for page in vblk do
12         if IS-VALID-DATA(page) then
13             data ← DO-2D-READ-SREQ(page);
14             DO-2D-WRITE-SREQ(data);
15 DO-2D-ERASE-SREQ(vblk);
```

advisable to redirect write sub-requests, since a 2D NAND chip itself employs *intra*-NAND work stealing to balance the sub-request queues for each NAND die. Hence, as derived in the algorithm, we trigger the *inter*-NAND write redirection (work stealing) only when the 2D NAND chip is idle, i.e., when we have no pending write requests in any 2D chips. Note that the sub-requests redirected from 3D to 2D NAND need to be *split* into multiple 2D page granularity sub-requests due to the distinct page granularities of 2D and 3D NAND, which is taken care of in line 5 of Algorithm 2.

5.2.2 Data migration

In Equation (2), if $v_{2D} > v_{3D}$, there is more percentage of valid data in the 2D NAND compared to the 3D NAND. As a result, the subsequent writes to the 2D NAND will trigger GC more frequently, causing potentially severe performance degradation and shortened lifetime. To avoid such scenarios, Centaur *pro-actively migrates* the data already present in 2D NAND to 3D NAND. However, the frequency of this migration plays a crucial role due to the 3D NAND "IDP unit." A page migrated prematurely from 2D NAND to 3D NAND will result in more write operations to 3D NAND. Owing to the out-of-place updates employed in NAND chips, such premature migration will cause the subsequent writes to cause a higher write-amplification [29] at the coarse granular "IDP units," thereby worsening the lifetime of 3D NAND. Hence, this premature migration is an undesirable artifact in Centaur. To avert this, we *postpone* the migration of data in 2D NAND to 3D NAND to the farthest point possible, i.e., the GC itself.

Another important aspect of migrating data from 2D to 3D NAND is the disparity in program-granularities between 2D NAND and 3D NAND. Just migrating one page from 2D to 3D NAND would result in small granularity writes to 3D NAND, thereby shortening its lifetime. Therefore, in Centaur, we propose multiple 2D pages to be *coalesced* into one "IDP unit" in 3D NAND while migrating.

Algorithm 3 represents the updated 2D NAND GC algorithm which migrates the coalesced 2D pages into a corresponding 3D NAND "IDP unit." Note that Algorithm 3 is described at a higher level, since GC involves all basic NAND operations, read, write and erase; hence, Do-2D/3D-write-sreq in line 9 and line 14 includes several FTL functions, such as the intra-NAND dispatcher and intra-/inter-NAND work stealers.

5.3 Write Indirection Caveat in Centaur

The request dispatch and write re-direction discussed in Sections 5.1 and 5.2.1 can result in the original data residing in one type of NAND, while the corresponding write request is being re-directed to another NAND type. More specifically, although the original data may reside in 3D NAND, the request dispatch or the write re-direction can cause the writes to be sent to 2D NAND. Depending on the granularity of the re-directed write, the other parts of the valid data in the same page may need to be fetched from a different NAND chip to ensure data integrity.

Let us assume as an example that data currently resides in 3D NAND, and 1 KB of the data are updated by a small write request. This write request will be processed by 2D NAND due to its small size. However, the 1 KB write request is typically smaller than the 2D page granularity, say 8KB. Therefore, the remaining 7KB data has to be fetched (read) from the data located in the 3D NAND chip to guarantee the “integrity" of the newly-written 2D NAND page. We want to emphasize that such inter-NAND data fetches are warranted to maintain the integrity of data in Centaur. Note also that, this is mainly for correctness, and we do not observe any significant impact on the write performance, whose breakdown can be found in Figure 14 in Section 6.2.1. Specifically, these additional read overheads are modeled in “2D Read" and “3D Read" in the write latency breakdown.

5.4 Overheads

Our Centaur design requires an extra 2D NAND mapping table apart from the 3D NAND mapping table. The size of this 2D NAND mapping table is larger than that of its 3D counterpart, since the 2D NAND chips employ smaller pages. To address this overhead, DFTL [28] can be adopted, and as a result, the mapping table is stored in the 2D NAND chips themselves. Thus, the overhead brought by the meta-data mapping table can be significantly reduced.

6 EVALUATION

6.1 Experimental Setup

To evaluate Centaur, our heterogeneous 2D/3D NAND SSD architecture, we augmented the SSDSim simulator [30] to model both 2D and 3D NAND chips. We used 12 write-dominated SSD traces [38] for our evaluations, and the important characteristics of these traces are given in Table 1. This table shows the number of read/write requests (in millions), the total amount of read/write data (in GBs), and the amount of uniquely-accessed read/write data (in GBs). Table 1 shows that our workload traces exhibit diverse characteristics.

In Centaur, to cover the entire spectrum of design space in terms of the heterogeneous NAND chip organization, we ran the following 3 configurations: (a) iso-capacity, (b) iso-die count (iso-area),

	reqs (in millions)		data (in GBs)		coverage (in GBs)	
trace	**read**	**write**	**read**	**write**	**read**	**write**
hm_1	0.58	0.028	8.476	0.553	1.53	1.11
prn_0	0.602	4.983	13.12	45.96	3.72	12.38
prn_1	8.464	2.77	181.35	30.78	73.78	11.52
proj_0	0.527	3.697	8.97	144.26	1.74	1.65
proj_1	21.143	2.497	750.36	25.57	693.5	9.03
proj_2	25.642	3.625	1015.9	168.68	409.37	155.13
prxy_0	0.384	12.135	3.04	53.8	0.19	0.25
src1_1	43.576	2.17	1485.6	30.34	116.69	4.16
src1_2	0.484	1.424	8.82	44.14	1.55	0.65
stg_1	1.4	0.796	79.52	5.98	79.42	0.39
usr_1	41.426	3.858	2079.2	56.12	651.16	24.56
usr_2	8.575	1.995	415.28	26.46	377.8	10.02

Table 1. Important characteristics of our workloads.

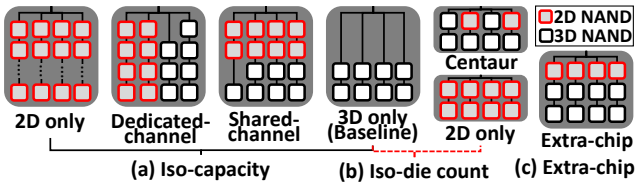


Fig. 10. Evaluated chip layouts and configurations.

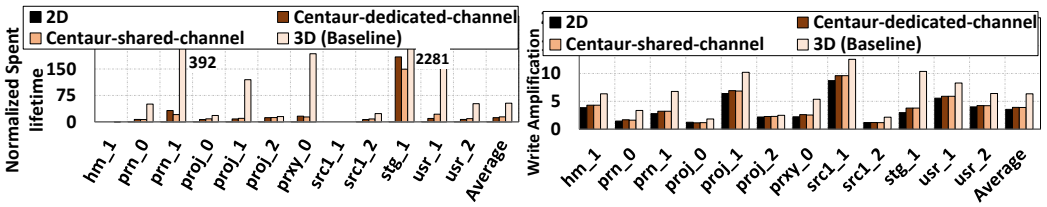
3D NAND chip parameters [36]	
(Read, Program, Erase)	(70μs, 900μs, 10ms)
(Die, Plane, Block, Page)	(1, 2, 1437, 768)
(Page size, Cell density)	(16KB, 3 bits per cell)
Program/Erase (PE) cycle	5000
(Chip capacity, IDP unit size)	(32 GB, 96KB)
2D NAND chip parameters [2]	
(Read, Program, Erase)	(35μs, 350μs, 1.5ms)
(Die, Plane, Block, Page)	(1, 2, 2048, 128)
(Page size, Cell density)	(8KB, 1 bit per cell)
Program/Erase (PE) cycle	60000
Chip capacity	4 GB
2D NAND chip parameters for extra-chip config [2]	
(Die, Plane, block, Page)	(1, 2, 1024 (512, 2048), 128)
Chip capacity	2 GB (1 GB, 4 GB)
SSD parameters	
FTL for both NAND type	Page-level & IDP-unit-level
Victim block selection	max# invalid pages
(GC trigger threshold, DRAM)	(5%, 64MB)
Transfer time per byte	5ns
(Over provision, Initial data)	(25%, 50%)
($DC_{Initial}$, DC_{Max}, DC_{Min})	(72KB, 88KB, 32KB)
$DC_{Adj\_threshold}$	1000
Intra-NAND wear-leveler	BET-based [23, 41]

Table 2. Characteristics of the evaluated SSDs.

and (c) extra-chip. In the iso-capacity configuration, the total capacity of the heterogeneous 2D/3D NAND SSD and those of the homogeneous 2D and 3D NAND SSDs are the same. Due to the capacity difference between the 2D and 3D NAND chips, *one* 3D NAND chip [36] is replaced with *eight* 2D NAND chips [2]. The iso-capacity configuration is the "fairest comparison" regarding the lifetime, even though the occupied areas by different configurations are different. This is because a larger capacity means more blocks and pages in an SSD, which decreases the GC frequency, thereby prolonging the SSD lifetime. Therefore, the iso-capacity comparisons are conducted to avoid such unfairness.

In the iso-die count comparison, the number of NAND chips in the heterogeneous and homogeneous SSDs are the same; but, the total capacities of these SSDs are different. This capacity disparity, as mentioned before, makes both lifetime and throughput comparisons unfair. However, Centaur can still outperform the baseline in most of the workloads tested.

The extra-chip configuration is used to conduct sensitivity tests on cases where the capacity of 2D NAND flash takes smaller shares (3% and 6%) of the overall SSD capacity. Note that the iso-capacity and iso-die configurations *cannot* be used to conduct such sensitivity test. This is because, changing the contribution of 2D NAND to the overall capacity also alters other factors, such as the total chip count and the overall capacity; hence, the extra-chip configuration is used in this sensitivity test. These three configurations are depicted in Figure 10. Note that, to perform a fair comparison, all these configurations share the *same number of channels*. The detailed parameters of these configurations are given in Table 2. Note also that the characteristics of our baseline 3D



(a) Spent-lifetime. (b) Write amplification.

Fig. 11. Spent-lifetime comparison under the iso-capacity configuration.

NAND SSD are similar to those of the state-of-the-art SSDs, such as Crucial BX300 240 GB [5], and Toshiba TR200 240 GB [6]; they also employ 8 32GB 3D dies.

6.2 Experimental Results

6.2.1 The iso-capacity comparison

The iso-capacity comparison is expected to reveal the lifetime and performance improvements brought by Centaur compared to a homogeneous 3D NAND-based SSD (baseline). Here, the results of two different NAND chip layouts, shared- and dedicated-channels, for proposed SSD are presented. The best iso-capacity setting, 2D NAND-based SSD, is shown as a "reference" to measure how well the proposed SSD fares in comparison.

Spent-lifetime: Figure 11a shows the spent-lifetime comparison, *normalized* to the case with a conventional 2D NAND-based SSD. The heterogeneous SSD spent-lifetime is calculated by the equation:

$$spent\text{-}lifetime = max(l_{2D},\ l_{3D}), \tag{3}$$

where the definitions of l_{2D} and l_{3D} are as in Equation (1). As can be observed from Figure 11a, the 2D NAND-based SSD performs the best, due to the small page granularity it employs and fewer GC copy operations it incurs, compared to the others. Also, both of our two heterogeneous SSD configurations outperform the conventional 3D NAND-based SSD. This is because the 2D NAND chips in the heterogeneous SSDs can execute the small writes, thereby not causing the write-amplification for the 3D NAND "IDP unit." It can also be observed that similar improvements are achieved by both the heterogeneous SSD configurations, since the dispatcher, work stealer and migrator in Centaur are independent of the underlying NAND chip layout. On average, the lifetime of the shared-channel heterogeneous SSD is $3.7x$ longer compared to the baseline.

Regarding the individual workloads, the spent-lifetime reductions on some workloads, such as prxy_0 and stg_1, are significantly higher than those of the other workloads. The reason is that the average write granularities of those workloads (shown in Table 1) are quite small (about 7KB); as a result, the spent-lifetime of these workloads is significantly degraded by these (small) writes. Hence, dispatching these small writes to the 2D NAND chips of Centaur can significantly reduce the amount of the spent-lifetime. In contrast, since the other workloads, such as proj_0 and proj_2, are not dominated by the small writes, the spent-lifetime of these workloads cannot be significantly reduced. Overall, Centaur can, in general, improve the SSD spent-lifetime of all workloads, and the spent-lifetime of small write-dominated workloads can be significantly reduced.

Figure 11b plots the write amplification (WA) comparison. The WA is defined as follows:

$$WA = \frac{\text{the total bytes being written}}{\text{the total bytes by write requests}}. \tag{4}$$

The WA values for the heterogeneous SSDs are close to those for the 2D NAND-based SSD and are much smaller than those for the 3D NAND-based SSD. This is because the 2D NAND chips in the

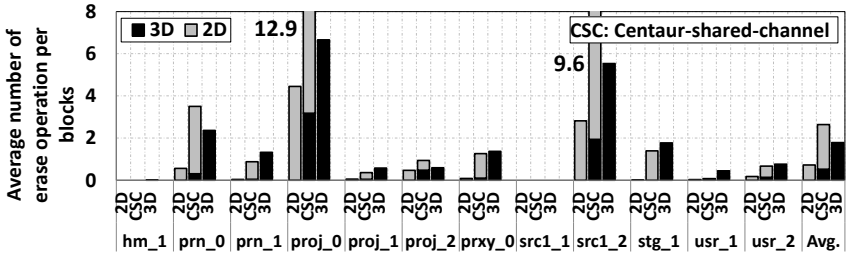
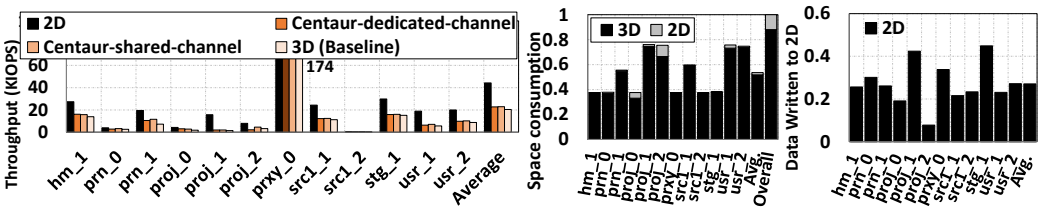


Fig. 12. The breakdown of erase operations under the iso-capacity configuration.

heterogeneous SSD reduce the potential write amplification caused by the small writes to the 3D NAND "IDP unit."

Figure 12 shows the average number of erase operations executed across blocks by each NAND type. As can be observed, in the heterogeneous SSDs, 2D NAND experiences more erase operations than 3D NAND. This is because, Centaur balances the total lifetime, *not* the average number of erase operations. In general, since the 2D SLC NAND lifetime is much longer than that of the 3D NAND, it can endure a larger number of erase operations. As a result, Centaur performs very well in terms of the overall lifetime.

Performance: The throughput improvements brought by our proposed SSD design are quantified in Figure 13a. As can be observed, both of our proposed SSDs perform better than the 3D-NAND based SSD in all the workloads tested, and the shared-channel one outperforms the dedicated-channel one. This is because the shared-channel SSD can process the requests in parallel. In contrast, the dedicated-channel heterogeneous SSD limits the channels being accessed in parallel across the 2D and 3D NAND chips. On average, the proposed SSD is 1.11x faster than the 3D NAND-based SSD, but 1.9x slower than the 2D NAND-based SSD, due to the high multi-chip parallelism exhibited by the 2D NAND SSD.



(a) Throughput. (b) Fraction of valid data on 2D/3D chips. (c) Fraction of data written to 2D chips.

Fig. 13. Comparisons under the iso-capacity configuration.

Final valid data and dispatched data of 2D/3D chips: Figure 13b plots the final valid data of 2D and 3D chips under the "iso-capacity" configuration. As can be observed, a majority of the data reside in 3D chips and 2D chips are not fully occupied. This is because, the total capacity of 2D chips is small and the GC overhead increases as the valid data that reside in chips increase. Hence, Centaur uses 2D chips "conservatively". Figure 13c shows the data dispatched to the 2D chips by the dispatcher. It can be observed that the amount of data sent to 2D chips varies across workloads, and the majority of the data are sent 3D chips. By doing so, Centaur can achieve a better SSD spent-lifetime.

Read/Write request breakdown: Figure 15 plots the breakdown of the read requests. The 2D NAND in Centaur can provide a limited latency reduction in both the proposed SSD cases, since

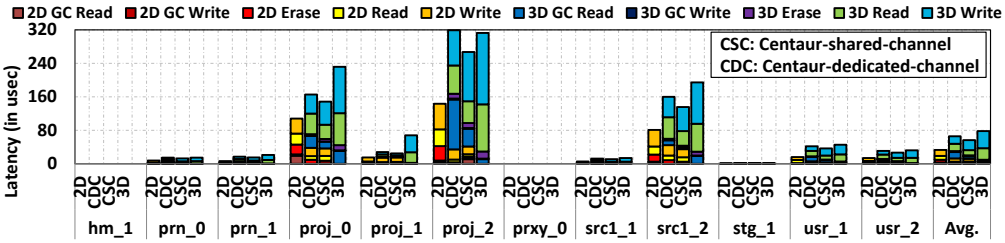


Fig. 14. Write request breakdown under the iso-capacity configuration.

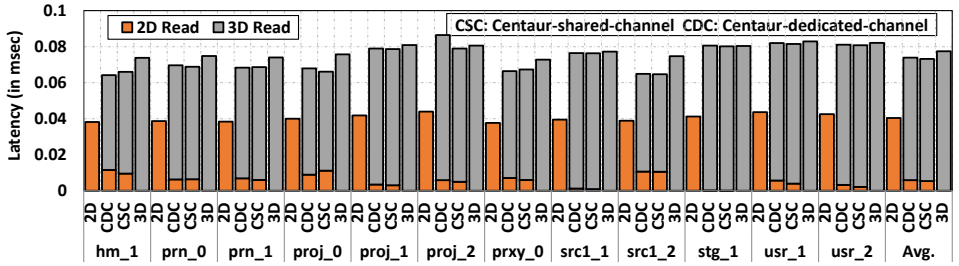
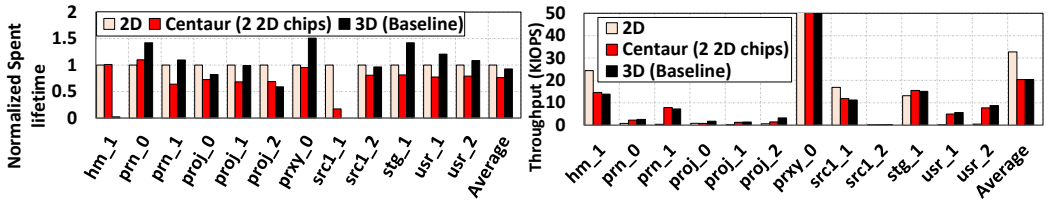


Fig. 15. Read request breakdown under the iso-capacity configuration.

the capacity of a 3D NAND chip is much larger than that of its 2D counterpart (consequently, most of the read requests are still served by 3D NAND).

Figure 14 plots the breakdown for the write requests. Note that since the write latencies of some workloads, such as hm_1 and stg_1, can be successfully hidden by the DRAM buffer, the write latencies of these workloads can be as low as 0. In comparison, the write latencies of the other workloads are reduced since the small writes are dispatched to the short-latency 2D chips. Another key observation is that the average write latencies of the channel-shared Centaur are shorter than those of the channel-dedicated Centaur. This is because the channel-shared Centaur can evenly distribute the writes across different channels; hence, the writes can be serviced by all channels simultaneously, thereby minimizing the write latencies. In contrast, the channel-dedicated Centaur can face the channel resource conflicts, where the writes are transferred to each chip one-by-one; as a result, the write latencies are longer than the channel-shared counterparts. Therefore, we propose to use the shared channel architecture for our Centaur.

6.2.2 The iso-die count comparison



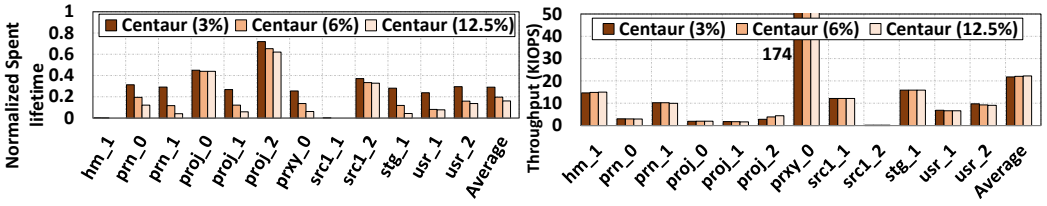
(a) Spent-lifetime. (b) Performance.

Fig. 16. Comparisons under the iso-die configuration.

Figures 16a and 16b plot, respectively, the lifetime and throughput comparison, under the iso-die-count configuration. Note that this comparison is *not* fair under both the metrics, since replacing a 3D chip with a 2D one reduces the overall SSD capacity significantly. This capacity reduction in turn increases the GC overheads, thereby reducing both lifetime and throughput. Hence, due to

the capacity reduction, Centaur performs badly under some workloads, such as hm_1 and proj_2, compared to the baseline. However, Centaur still outperforms the baseline in most of the workloads.

6.2.3 The extra-chip comparison



(a) Spent-lifetime. (b) Performance.

Fig. 17. Comparisons with Different contributions of 2D capacity under the extra-chip configuration.

Since the previous two comparisons cannot fairly dem-onstrate the impact of Centaur under different contributions of the 2D NAND to the total capacity, we also conducted an extra-chip comparison. Figures 17a and 17b plot, respectively, the lifetime and throughput comparisons with three different contributions of 2D NAND to the total capacity: 3%, 6%, and 12.6%. For example, 3% means that the capacity of the total 2D NAND chips accounts for 3% of the total SSD capacity. We observe that, as this percentage increases, both lifetime and throughput are improved significantly.

6.2.4 DRAM capacity sensitivity results

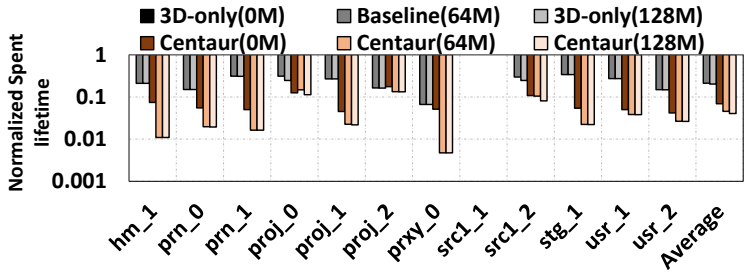


Fig. 18. Sensitivity test for spent-lifetime across different amount of DRAM capacities under the iso-capacity configuration. (Normalized to Baseline(0M))

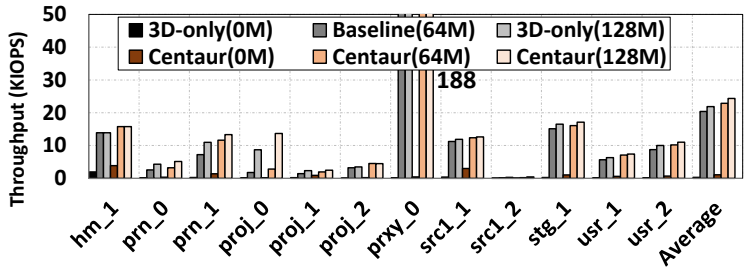
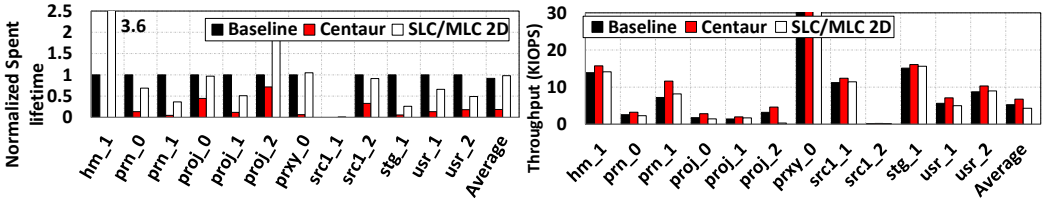


Fig. 19. Sensitivity test for performance across different amount of DRAM capacities under the iso-capacity configuration.

Figures 18 and 19 show the lifetime and throughput comparison under these different DRAM capacities: 0M, 64M (baseline), and 128M. It is can be observed from Figure 18 that, as the DRAM

capacity of the baseline increases from 0M to 64M, the lifetime of most workloads is significantly reduced. This is because the small writes with a high temporal locality are successfully aggregated to reduce the spent-lifetime. However, increasing the amount of DRAM buffer from 64M to 128M only slightly reduces the spent-lifetime. This is because SSDs periodically (30 seconds in our experiments) receive data flush commands from file systems and databases to flush data from its DRAM buffers to NAND flash. As a result, DRAM buffer *cannot* accumulate small writes until an "IDP unit"-granularity is reached, before writing it into 3D NAND. In contrast, Centaur can further reduce the spent-lifetime under the 64M DRAM buffer compared to the baseline. Regarding the throughput, as shown in Figure 19, the throughput of both the baseline and Centaur can be improved by a larger DRAM capacity; but, the improvements quickly saturates, as the DRAM capacity is doubled from 64M to 128MB. In summary, Centaur can improve both the degraded spent-lifetime and throughput of 3D NAND-based SSDs under different amount of DRAM capacities.

6.2.5 Comparison against an SLC/MLC NAND SSD



(a) Spent-lifetime. (b) Performance.

Fig. 20. Centaur and the approach proposed in [22] under the iso-capacity configuration.

We are *not* aware of any prior works, which propose to integrate 2D NAND and 3D NAND in the same SSD to address the lifetime/performance problem of 3D NAND. The SLC/MLC SSD [22] combines SLC and MLC NAND in the *same* monolithic 2D SSD. The SLC/MLC SSD dispatches the hot and cold data to SLC and MLC NAND, respectively. The hotness information is calculated using the request granularity distribution of the last epoch. To be more specific, the k-means (k equals 2 in this case) clustering algorithm is used to cluster the distribution into two parts: hot and cold granularities. The clustering thresholds are typically 4KB or 8KB, which means that, if a request is smaller than or equal to 4KB or 8KB, they are dispatched to SLC NAND; otherwise, to MLC NAND. To prevent SLC NAND from wearing out faster than MLC NAND, some hot requests may be dispatched to MLC NAND according to the lifetime difference between SLC NAND and MLC NAND. However, since the SLC/MLC SSD is essentially a 2D SSD proposal, this prior work is *not* aware of distinct program granularities between 2D and 3D NAND. As a result, it cannot address the targeted 3D NAND performance and lifetime issues.

Clearly, the SLC/MLC SSD cannot be compared against Centaur directly without modification, since it is optimized for improving the throughput of a dual-channel memory-constrained 2D SSD, where SLC NAND and MLC NAND use distinct channels. As a result, to perform a comparison between Centaur and the SLC/MLC 2D SSD, we replace their memory-optimized specially-designed block-level FTL with Centaur's page-level FTL, so that their dispatch management can be run on top of the conventional multi-channel SSD. In our implementation of [22] in a 2D/3D heterogeneous SSD, we mapped 2D NAND to SLC NAND and 3D NAND to MLC NAND.

Figures 20a and 20b show the lifetime and throughput results for Centaur compared to the hybrid SLC/MLC 2D SSD proposed by [22]. *Centaur outperforms the SLC/MLC 2D SSD in both the metrics*, owing to our optimization for 3D NAND "IDP unit." It is to be noted that the SLC/MLC 2D SSD is agnostic to the 3D NAND "IDP unit"; so, the small writes (which are larger than 8KB but much

smaller than 96KB) are still executed by the 3D NAND. As a result, the 3D NAND lifetime in the SLC/MLC 2D SSD configuration becomes much shorter compared to that in Centaur. Figures 20a and 20b also compare our Centaur against the IDP-disabled (IDP-less) hybrid SLC/MLC 2D SSD (the last bar for each benchmark). These results show that disabling the IDP can eliminate the small write problem, but then the hybrid SLC/MLC SSD can suffer a very low throughput. The outcome of this evaluation is that, one should not simply pick an existing hybrid dispatching strategy and use it, as is, in the 2D/3D heterogeneous SSD. Instead, as Centaur does, one needs to be aware of the 3D NAND-specific characteristics.

6.2.6 Centaur inter-NAND dispatcher results

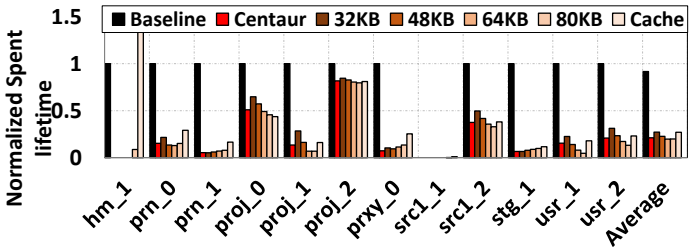


Fig. 21. Sensitivity test for spent-lifetime across different DCs under the iso-capacity configuration.

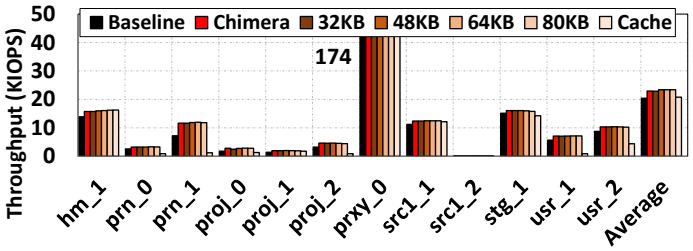


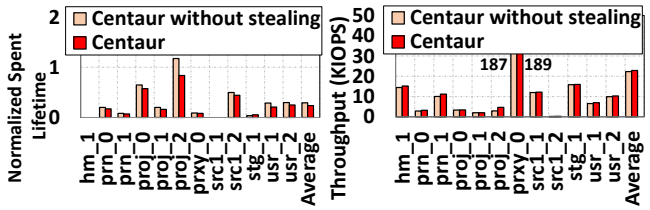
Fig. 22. Sensitivity test for performance across different DCs under the iso-capacity configuration.

We now report results from a dispatch condition (DC) sensitivity test of the shared-channel case, under the iso-capacity configuration. Recall that DC is the condition used (by the dispatcher) to dispatch requests to 2D and 3D NAND. Figures 21 and 22 give the lifetime and performance results, respectively, for the baseline, cache mode, Centaur and Centaur with the static DC value, under the iso-capacity configuration. The best static DC values for the lifetime for each trace fall in the range between 32KB and cache mode (96K). This is because the request granularity distributions of the different workloads are different; hence, Centaur with the static DC value *cannot* achieve the best lifetime or performance. Note that, as shown in Figure 21, cache mode, which is essentially a special case of Centaur with the DC value set to 96K, shortens the lifetime of the 2D NAND chips, due to the 2D/3D redundant processing problem mentioned in Section 4.

Note that the best static DC value for the throughput is *not* the same as the best static DC value for the lifetime, since Centaur is mainly optimized for the lifetime. Also, note that, cache mode, which is discussed in Section 4 and Figure 7b-1, is essentially a special case of Centaur, where the DC is set to 96K.

6.2.7 Centaur inter-NAND work stealer results

Figures 23a and 23b plot the Centaur lifetime and throughput results, respectively, *with* and *without* the inter-NAND work stealer optimization enabled. On average, Centaur with the inter-NAND

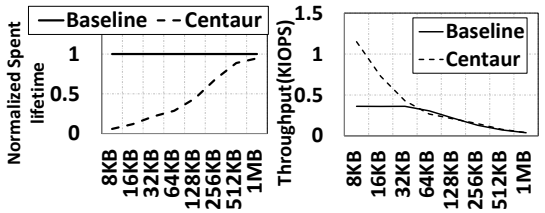


(a) Spent-lifetime. (b) Performance.

Fig. 23. Comparisons with and without using the workload stealing technique.

work stealer enabled improves lifetime by 13%. In particular, workloads such as *proj_2* benefit significantly, owing to the fewer number of small writes. This is because the 3D NAND chips are replaced by the same capacity 2D NAND chips under the iso-capacity configuration. Consequently, with the same number of writes to the 3D NAND, the 3D NAND chips in the proposed SSDs would be under a heavier workload compared to the baseline. This heavier workload triggers more GC operations, which in turn shortens the lifetime of the 3D NAND chips. Therefore, to prevent this lifetime problem, the existence of the inter-NAND work stealer in Centaur is *critical*, as it balances the writes across the 2D and 3D chips.

6.2.8 Centaur data migration results



(a) Spent-lifetime. (b) Performance.

Fig. 24. Comparisons with uni-size data under the iso-capacity configuration.

The data migration in Centaur *cannot* be disabled, since it is used to guarantee that the 2D NAND chips never run out of capacity. As a result, to isolate the performance and lifetime improvements brought by data migration, we disable the dispatcher *and* the inter-NAND work stealer. Note that the dispatcher is disabled by evaluating Centaur with the fixed granularity random pattern write requests. Figures 24a and 24b plot the lifetime and throughput results, respectively, for the iso-capacity configuration, with the dispatcher disabled. Both the performance and lifetime for the small granularity requests are significantly improved, since all the requests are executed by the 2D NAND, and the lifetime of the 3D NAND is prolonged. However, the large granularity requests experience only limited or no lifetime improvement, since they encounter the same lifetime issue discussed in Section 6.2.7. In Centaur, this lifetime issue is addressed by employing an inter-NAND work stealer.

7 RELATED WORK

We discuss related work under four categories:

Dynamic page allocation proposals (intra-NAND work stealers): The dynamic(-granularity) page allocation algorithms [30, 53, 54] are proposed to address the drawbacks of the static page allocation (intra-NAND dispatcher) algorithms [35], where a request can only be processed by

the specified NAND plane based on its address. Note that this constraint of static page allocation algorithms can result in workload imbalance. And, this imbalance can in turn degrade the overall SSD performance significantly. To improve performance, the dynamic page allocation algorithms allow the lightly-loaded NAND planes to steal the requests originally destined to the heavily-loaded NAND planes. However, these algorithms are designed for the homogeneous NAND SSDs, and as shown in Section 6.2.7, they are *not* optimal for 2D/3D heterogeneous SSD, as far as lifetime is concerned. Instead, in Centaur, the lifetime is successfully optimized by the inter-NAND work stealer.

Hybrid-drive solutions: We are *not* aware of any 2D/3D hybrid-drive solution, but a few SSD/HDD hybrid-drive solutions are available in Linux, such as Bcache [7], dm-cache [10], and Flashcache [12]. Such prior solutions are "performance-driven," where random and sequential writes are dispatched to SSDs and HDDs, respectively; hence, the system can extract the best performance benefits from both types of drives. However, these prior solutions are *not* lifetime-optimized when considering the 2D/3D hybrids. That is, under such solutions, small random writes would still be sent to 3D SSDs under the employed load-balancing mechanisms [7]. Further, the hybrid-drive solutions treat (2D) SSDs as an additional write-back cache; so, the capacity of (2D) SSDs is not counted in the total capacity. In comparison, Centaur is optimized primarily for the SSD lifetime, and *both* 2D and 3D NAND chips are accounted towards the total storage capacity.

Large page granularity proposals: Superpage [19] is a proposal to combine the pages in multiple planes, dies, chips and even channels to form a "superpage." With the superpage support, the SSD throughput can be improved, and the FTL design becomes simpler. However, an SSD with a (large) superpage can face the same lifetime and performance issues discussed before. To address these issues, the granularity of the "superpage" can be reduced by combining fewer pages. However, as discussed earlier, the FSP page granularity *cannot* be arbitrarily reduced. Therefore, in the IDP-enabled 3D NAND SSDs, one would still need Centaur to address the lifetime issues caused by the small writes.

8 CONCLUSION

In this paper, we propose Centaur, a novel 2D/3D NAND heterogeneous SSD to address the lifetime and performance degradation problems caused by the "IDP unit" write granularity in emerging 3D NAND SSDs. Specifically, as part of Centaur, we propose: (a) a lifetime-aware inter-NAND request dispatcher, (b) a lifetime-aware inter-NAND work stealer, and (c) a data migration strategy. Our dispatcher dispatches the write requests according to a dynamically-adjustable dispatch condition, to prevent the 3D NAND chips from executing small granularity write requests. On the other hand, the inter-NAND work stealer and the data migration components together *balance* the lifetime between the 2D and 3D NAND chips. Our extensive experiments with various workloads show that the lifetime and performance of a state-of-the-art 3D NAND based SSD are improved by $3.7x$ and $1.11x$, respectively, under the iso-capacity configuration.

ACKNOWLEDGMENTS

This research is supported by NSF grants 1439021, 1439057, 1409095, 1626251, 1629915, 1629129, 1526750 and 1908793, and a grant from Intel. Dr. Jung is supported in part by NRF 2016R1C1B2015312, DOE DE-AC02-05CH 11231, NRF-2015M3C4A7065645, KAIST Start-Up Grant (G01190015), and MemRay grant (G01190170). AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

REFERENCES

[1] 200. Samsung K9XXG08UXD datasheet. https://www.samsung.com/. (March 200).
[2] 2007. Micron MT29F8G08BAA datasheet. https://www.micron.com/products/nand-flash/. (Feb. 2007).
[3] 2014. ONFI 4.0 Specification. http://www.onfi.org/. (April 2014).
[4] 2016. Samsung 960 EVO SSD. http://www.samsung.com/semiconductor/minisite/ssd/product/consumer/960evo.html. (Sept. 2016).
[5] 2017. Crucial BX300 SSD. http://www.crucial.com/usa/en/storage-ssd-bx300. (Aug. 2017).
[6] 2017. Toshiba TR200 3D NAND SSD. https://www.ocz.com/us/ssd/tr200. (Aug. 2017).
[7] 2018. Bcache: Linux kernel block layer cache. https://bcache.evilpiepirate.org/. (March 2018).
[8] 2018. Berkeley DB. https://www.oracle.com/database/berkeley-db/db.html. (2018).
[9] 2018. btrfs file-system. https://btrfs.wiki.kernel.org/index.php/Main_Page. (2018).
[10] 2018. dm-cache: a Linux kernel's device mapper target that allows creation of hybrid volumes. https://en.wikipedia.org/wiki/Dm-cache. (Oct. 2018).
[11] 2018. ext4 file-system. https://en.wikipedia.org/wiki/Ext4. (2018).
[12] 2018. flashcache: a disk cache component for the Linux kernel. https://en.wikipedia.org/wiki/Flashcache. (Oct. 2018).
[13] 2020. Apple iPhone. https://www.apple.com/iphone/. (April 2020).
[14] Jacob Alter, Ji Xue, Alma Dimnaku, and Evgenia Smirni. 2019. SSD Failures in the Field: Symptoms, Causes, and Prediction Models *(SC âĂŹ19)*.
[15] Seiichi Aritome. 2015. *NAND Flash Memory Technologies*. John Wiley & Sons, Inc.
[16] Rodolfo Azevedo, John D. Davis, Karin Strauss, Parikshit Gopalan, Mark Manasse, and Sergey Yekhanin. 2013. Zombie Memory: Extending Memory Lifetime by Reviving Dead Blocks. In *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA)*.
[17] A. Ban. 1995. Flash file system. https://www.google.com/patents/US5404485. (April 4 1995). US Patent 5,404,485.
[18] Y. Cai, S. Ghose, Y. Luo, K. Mai, O. Mutlu, and E. F. Haratsch. 2017. Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*.
[19] Adrian M. Caulfield, Laura M. Grupp, and Steven Swanson. 2009. Gordon: Using Flash Memory to Build Fast, Power-efficient Clusters for Data-intensive Applications. In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.
[20] Adrian M. Caulfield and Steven Swanson. 2013. QuickSAN: A Storage Area Network for Fast, Distributed, Solid State Disks. In *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA)*.
[21] R. A. Cernea, L. Pham, F. Moogat, S. Chan, B. Le, Y. Li, S. Tsao, T. Y. Tseng, K. Nguyen, J. Li, J. Hu, J. H. Yuh, C. Hsu, F. Zhang, T. Kamei, H. Nasu, P. Kliza, K. Htoo, J. Lutze, Y. Dong, M. Higashitani, J. Yang, H. S. Lin, V. Sakhamuri, A. Li, F. Pan, S. Yadala, S. Taigor, K. Pradhan, J. Lan, J. Chan, T. Abe, Y. Fukuda, H. Mukai, K. Kawakami, C. Liang, T. Ip, S. F. Chang, J. Lakshmipathi, S. Huynh, D. Pantelakis, M. Mofidi, and K. Quader. 2009. A 34 MB/s MLC Write Throughput 16 Gb NAND With All Bit Line Architecture on 56 nm Technology. *IEEE Journal of Solid-State Circuits*.
[22] L. P. Chang. 2010. A Hybrid Approach to NAND-Flash-Based Solid-State Disks. *IEEE Trans. Comput.*
[23] Y. H. Chang, J. W. Hsieh, and T. W. Kuo. 2010. Improving Flash Wear-Leveling by Proactively Moving Static Data. *IEEE Trans. Comput.*
[24] Chih-Ping Chen, H. T. Lue, Chih-Chang Hsieh, Kuo-Pin Chang, K. Y. Hsieh, and C. Y. Lu. 2010. Study of fast initial charge loss and it's impact on the programmed states Vt distribution of charge-trapping NAND Flash. In *2010 International Electron Devices Meeting*.
[25] F. Chen, M. Yang, Y. Chang, and T. Kuo. 2015. PWL: A progressive wear leveling to minimize data migration overheads for NAND flash devices. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*.
[26] C. Monzio Compagnoni, A. Goda, A. S. Spinelli, P. Feeley, A. L. Lacaita, and A. Visconti. 2017. Reviewing the Evolution of the NAND Flash Technology. *Proc. IEEE*.
[27] Cagdas Dirik and Bruce Jacob. 2009. The Performance of PC Solid-state Disks (SSDs) As a Function of Bandwidth, Concurrency, Device Architecture, and System Organization. In *Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA)*.
[28] Aayush Gupta, Youngjae Kim, and Bhuvan Urgaonkar. 2009. DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings. In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.
[29] Xiao-Yu Hu, Evangelos Eleftheriou, Robert Haas, Ilias Iliadis, and Roman Pletka. 2009. Write Amplification Analysis in Flash-based Solid State Drives. In *Proceedings of SYSTOR 2009*.
[30] Yang Hu, Hong Jiang, Dan Feng, Lei Tian, Hao Luo, and Shuping Zhang. 2011. Performance impact and interplay of SSD parallelism through advanced commands, allocation strategy and data granularity. In *Proceedings of the international conference on Supercomputing (SC)*.

[31] Jian Huang, Anirudh Badam, Moinuddin K. Qureshi, and Karsten Schwan. 2015. Unified Address Translation for Memory-mapped SSDs with FlashMap. In *Proceedings of the 42Nd Annual International Symposium on Computer Architecture (ISCA)*.

[32] Jae-Woo Im, Woo-Pyo Jeong, Doo-Hyun Kim, Sang-Wan Nam, Dong-Kyo Shim, Myung-Hoon Choi, Hyun-Jun Yoon, Dae-Han Kim, You-Se Kim, Hyun-Wook Park, and others. 2015. 7.2 A 128Gb 3b/cell V-NAND flash memory with 1Gb/s I/O rate. In *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*.

[33] Manzur Gill Joe E. Brewer. 2007. *NonVolatile Memory Technologies with Emphasis on Flash*. John Wiley & Sons, Inc.

[34] Dawoon Jung, Jeong-UK Kang, Heeseung Jo, Jin-Soo Kim, and Joonwon Lee. 2010. Superblock FTL: A superblock-based flash translation layer with a hybrid address translation scheme. *ACM Transactions on Embedded Computing Systems*.

[35] Myoungsoo Jung and Mahmut Kandemir. 2012. An Evaluation of Different Page Allocation Strategies on High-speed SSDs. In *Proceedings of the 4th USENIX Conference on Hot Topics in Storage and File Systems (HotStorage)*.

[36] D. Kang, W. Jeong, C. Kim, D. H. Kim, Y. S. Cho, K. T. Kang, J. Ryu, K. M. Kang, S. Lee, W. Kim, H. Lee, J. Yu, N. Choi, D. S. Jang, J. D. Ihm, D. Kim, Y. S. Min, M. S. Kim, A. S. Park, J. I. Son, I. M. Kim, P. Kwak, B. K. Jung, D. S. Lee, H. Kim, H. J. Yang, D. S. Byeon, K. T. Park, K. H. Kyung, and J. H. Choi. 2016. 7.1 256Gb 3b/cell V-NAND flash memory with 48 stacked WL layers. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*.

[37] C. Kim, J. H. Cho, W. Jeong, I. h Park, H. W. Park, D. H. Kim, D. Kang, S. Lee, J. S. Lee, W. Kim, J. Park, Y. l Ahn, J. Lee, J. h Lee, S. Kim, H. J. Yoon, J. Yu, N. Choi, Y. Kwon, N. Kim, H. Jang, J. Park, S. Song, Y. Park, J. Bang, S. Hong, B. Jeong, H. J. Kim, C. Lee, Y. S. Min, I. Lee, I. M. Kim, S. H. Kim, D. Yoon, K. S. Kim, Y. Choi, M. Kim, H. Kim, P. Kwak, J. D. Ihm, D. S. Byeon, J. y Lee, K. T. Park, and K. h Kyung. 2017. 11.4 A 512Gb 3b/cell 64-stacked WL 3D V-NAND flash memory. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*.

[38] Miryeong Kwon, Jie Zhang, Gyuyoung Park, Wonil Choi, David Donofrio, John Shalf, Mahmut Kandemir, and Myoungsoo Jung. 2017. TraceTracker: Hardware/Software Co-Evaluation for Large-Scale I/O Workload Reconstruction. In *2016 IEEE International Symposium on Workload Characterization (IISWC)*.

[39] Sang-Won Lee, Dong-Joo Park, Tae-Sun Chung, Dong-Ho Lee, Sangwon Park, and Ha-Joo Song. 2007. A Log Buffer-based Flash Translation Layer Using Fully-associative Sector Translation. *ACM Trans. Embed. Comput. Syst.*.

[40] Cheng Li, Philip Shilane, Fred Douglis, Hyong Shim, Stephen Smaldone, and Grant Wallace. 2014. Nitro: A Capacity-Optimized SSD Cache for Primary Storage. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*.

[41] J. Li, X. Xu, X. Peng, and J. Liao. 2019. Pattern-based Write Scheduling and Read Balance-oriented Wear-Leveling for Solid State Drivers. In *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*.

[42] Yongkun Li, Patrick P.C. Lee, and John C.S. Lui. 2013. Stochastic Modeling of Large-scale Solid-state Storage Systems: Analysis, Design Tradeoffs and Optimization. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*.

[43] Chunyi Liu, Jagadish Kotra, Myoungsoo Jung, and Mahmut Kandemir. 2018. PEN: Design and Evaluation of Partial-Erase for 3D NAND-Based High Density SSDs. In *16th USENIX Conference on File and Storage Technologies (FAST 18)*.

[44] Y. Luo, S. Ghose, Y. Cai, E. F. Haratsch, and O. Mutlu. 2018. HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness. In *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*.

[45] Rino Micheloni, Luca Crippa, and Roberto Ravasio. 2007. Double page programming system and method. https://www.google.tl/patents/US20070030732. (Feb. 8 2007). US Patent 20070030732.

[46] M. Murugan and D. H. C. Du. 2011. Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead. In *2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST)*.

[47] K. Park, O. Kwon, S. Yoon, M. Choi, I. Kim, B. Kim, M. Kim, Y. Choi, S. Shin, Y. Song, J. Park, J. Lee, C. Eun, H. Lee, H. Kim, J. Lee, J. Kim, T. Kweon, H. Yoon, T. Kim, D. Shim, J. Sel, J. Shin, P. Kwak, J. Han, K. Kim, S. Lee, Y. Lim, and T. Jung. 2011. A 7MB/s 64Gb 3-bit/cell DDR NAND flash memory in 20nm-node technology. In *2011 IEEE International Solid-State Circuits Conference (ISSCC)*.

[48] Alessia Marelli Rino Micheloni, Luca Crippa. 2010. *Inside NAND Flash Memory*. Springer Netherlands.

[49] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. 2016. Flash Reliability in Production: The Expected and the Unexpected. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*.

[50] N. Shibata, K. Kanda, T. Shimizu, J. Nakai, O. Nagao, N. Kobayashi, M. Miakashi, Y. Nagadomi, T. Nakano, T. Kawabe, T. Shibuya, M. Sako, K. Yanagidaira, T. Hashimoto, H. Date, M. Sato, T. Nakagawa, H. Takamoto, J. Musha, T. Minamoto, M. Uda, D. Nakamura, K. Sakurai, T. Yamashita, J. Zhou, R. Tachibana, T. Takagiwa, T. Sugimoto, M. Ogawa, Y. Ochi, K. Kawaguchi, M. Kojima, T. Ogawa, T. Hashiguchi, R. Fukuda, M. Masuda, K. Kawakami, T. Someya, Y. Kajitani, Y. Matsumoto, N. Morozumi, J. Sato, N. Raghunathan, Y. L. Koh, S. Chen, J. Lee, H. Nasu, H. Sugawara, K. Hosono, T. Hisada, T. Kaneko, and H. Nakamura. 2019. 13.1 A 1.33Tb 4-bit/Cell 3D-Flash Memory on a 96-Word-Line-Layer Technology. In *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*.

[51] K. S. Shim, E. S. Choi, S. W. Jung, S. H. Kim, H. S. Yoo, K. S. Jeon, H. S. Joo, J. S. Oh, Y. S. Jang, K. J. Park, S. M. Choi, S. B. Lee, J. D. Koh, K. H. Lee, J. Y. Lee, S. H. Oh, S. H. Pyi, G. S. Cho, S. K. Park, J. W. Kim, S. K. Lee, and S. J. Hong. 2012. Inherent Issues and Challenges of Program Disturbance of 3D NAND Flash Cell. In *2012 4th IEEE International Memory Workshop*.

[52] Tomoharu Tanaka, Mark Helm, Tommaso Vali, Ramin Ghodsi, Koichi Kawai, Jae-Kwan Park, Shigekazu Yamada, Feng Pan, Yuichi Einaga, Ali Ghalam, and others. 2016. 7.7 A 768Gb 3b/cell 3D-floating-gate NAND flash memory. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*.

[53] Arash Tavakkol, Mohammad Arjomand, and Hamid Sarbazi-Azad. 2014. Unleashing the Potentials of Dynamism for Page Allocation Strategies in SSDs. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*.

[54] Arash Tavakkol, Pooyan Mehrvarzy, Mohammad Arjomand, and Hamid Sarbazi-Azad. 2016. Performance Evaluation of Dynamic Page Allocation Strategies in SSDs. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*.

[55] Benny Van Houdt. 2013. A Mean Field Model for a Class of Garbage Collection Algorithms in Flash-based Solid State Drives. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*.

[56] R. Yamashita, S. Magia, T. Higuchi, K. Yoneya, T. Yamamura, H. Mizukoshi, S. Zaitsu, M. Yamashita, S. Toyama, N. Kamae, J. Lee, S. Chen, J. Tao, W. Mak, X. Zhang, Y. Yu, Y. Utsunomiya, Y. Kato, M. Sakai, M. Matsumoto, H. Chibvongodze, N. Ookuma, H. Yabe, S. Taigor, R. Samineni, T. Kodama, Y. Kamata, Y. Namai, J. Huynh, S. E. Wang, Y. He, T. Pham, V. Saraf, A. Petkar, M. Watanabe, K. Hayashi, P. Swarnkar, H. Miwa, A. Pradhan, S. Dey, D. Dwibedy, T. Xavier, M. Balaga, S. Agarwal, S. Kulkarni, Z. Papasaheb, S. Deora, P. Hong, M. Wei, G. Balakrishnan, T. Ariki, K. Verma, C. Siau, Y. Dong, C. H. Lu, T. Miwa, and F. Moogat. 2017. 11.1 A 512Gb 3b/cell flash memory on 64-word-line-layer BiCS technology. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*.

[57] M. C. Yang, Y. M. Chang, C. W. Tsao, P. C. Huang, Y. H. Chang, and T. W. Kuo. 2014. Garbage collection and wear leveling for flash memory: Past and future. In *2014 International Conference on Smart Computing (SMARTCOMP)*.

[58] Q. Yang and J. Ren. 2011. I-CASH: Intelligently Coupled Array of SSD and HDD. In *2011 IEEE 17th International Symposium on High Performance Computer Architecture (HPCA)*.

Received January 2020; revised February 2020; accepted March 2020