The Effect of Host Data Patterns on SSD Write Performance

Doug Rollins, Senior Strategic Applications Engineer

Micron Technology, Inc.

Technical Marketing Brief

About SSDs

Solid state drives (SSDs) are storage devices that have generated substantial interest in recent years for their low power consumption, higher performance, and drop-in HDD-replacement capability.

Despite their deceptively similar appearance, SSDs are not functionally identical to rotating drives (HDDs). This technical marketing brief focuses on one key way in which these devices differ: the effect that traffic patterns have on write performance.

How SSDs Are Built

This brief is limited to examining SSDs that use NAND Flash as the basic storage media. NAND-based SSDs are organized by ***blocks*** and ***pages***.

The smallest element that can be written on an SSD is a page, whose size is dependent on the underlying NAND design. Common page sizes are 4K and 8K; this means that SSDs have a write granularity of 4K or 8K. When new data is written, it is written one page at a time.

NAND blocks are constructed of groups of pages. Block and page sizes vary with the NAND design. For this discussion, we will assume that one block contains 128 pages and that each page is 4K. Within an SSD, a block is the smallest unit of NAND that can be erased. This means that when erasing NAND in preparation for writing new data, the NAND must be erased one block at a time.

Note that actual block and page sizes may vary depending on the media used.

Simplified SSD Diagram

Figure 1 shows a very simplified diagram of a NAND-based SSD. Each blue box represents a single NAND page, while each vertical column represents a block. The exact amount of data that can be stored on a given page or block is

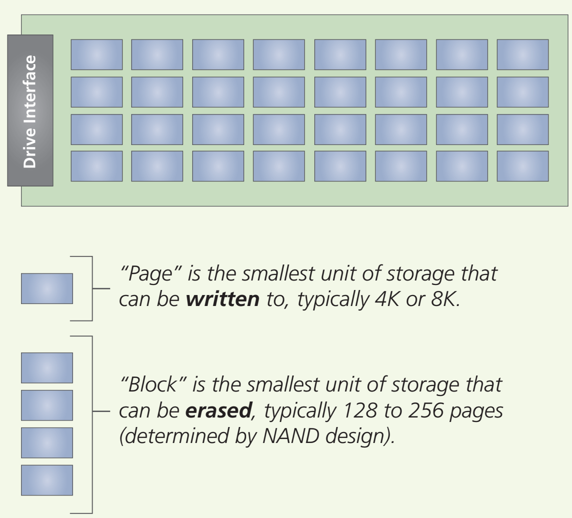


Figure 1: Block and page structure in NAND Flash-based SSDs

primarily a function of the design of the underlying media, but page sizes of 4K and 8K are common, as are block sizes of 128 or 256 pages.

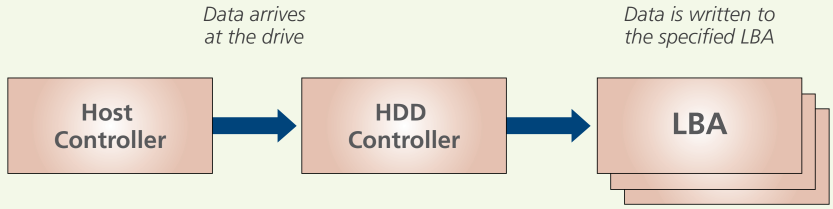


Figure 2: Rotating drive write scheme on a full drive—new data simply overwrites old data in a single step.

Unlike rotating drives, the NAND storage cells on a full SSD (an SSD with data already stored in its NAND cells, but possibly invalidated, or "deleted" by the host) must be erased before they can be written to. Rotating drives can do this in a single step (simply overwriting the existing, invalidated data), as shown in Figure 2.

But because NAND that contains existing data must be erased before it can be written to, an SSD must perform two functions to store new data in cells containing old (invalidated) data: first the cell must be erased, and then it can be written to, as shown in Figure 3.

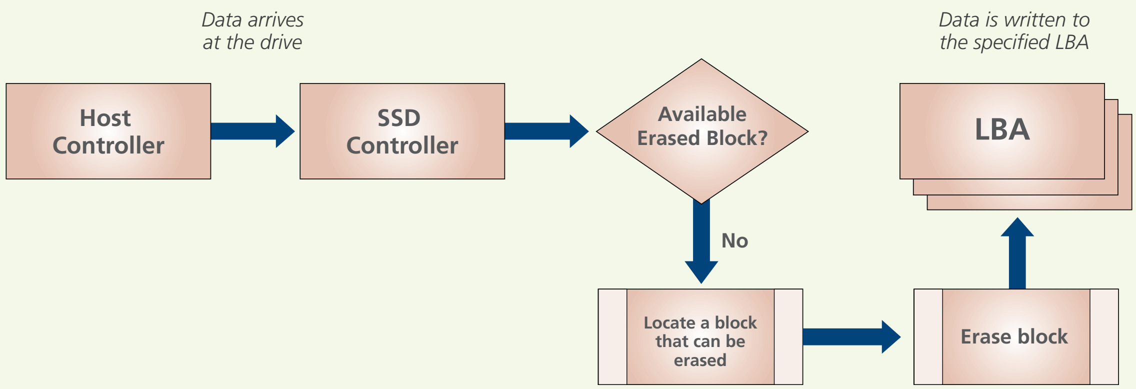


Figure 3: Two-step write process on a full drive—erase, then write. (Additional details are available in the Micron technical marketing brief "Media Management in Solid State Drives: Multistep WRITE Operations.")

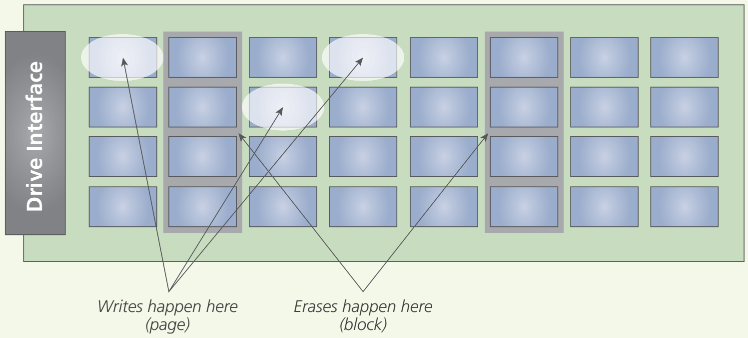


Figure 4: How data is erased and written in blocks and pages.

Looking back at the simplified SSD diagram in Figure 1, there is a disparity between the smallest amount of data that can be written and the smallest amount that can be erased (in preparation for the incoming data), as shown in Figure 4.

Effect of Page Write/Block Erase Disparity: Garbage Collection

In order to prepare for new, incoming data (by erasing invalid data blocks), the valid data must be separated from invalid data and moved to an unoccupied storage location, so that the block (that now contains only invalid data) can be erased. This process of gathering valid data, moving it, and erasing the block is known as garbage collection. For simplicity, the following examples will focus on a 4K write transfer.

The Effect of Host Patterns

Write performance is heavily influenced by host data patterns; large numbers of small random transfers decrease write performance, while large numbers of sequential transfers do so to a much smaller degree.

This difference is due to data-invalidation patterns, which are a function of how the data is written and the pattern that follows when the data is invalidated (marked for deletion).

Random, small-block patterns

In the example shown in Figure 5, an empty SSD is filled one page at a time (with 4K random data) until it is full. As in Figure 4, blue boxes represent empty pages, while yellow boxes with a D in the center represent pages with valid data.

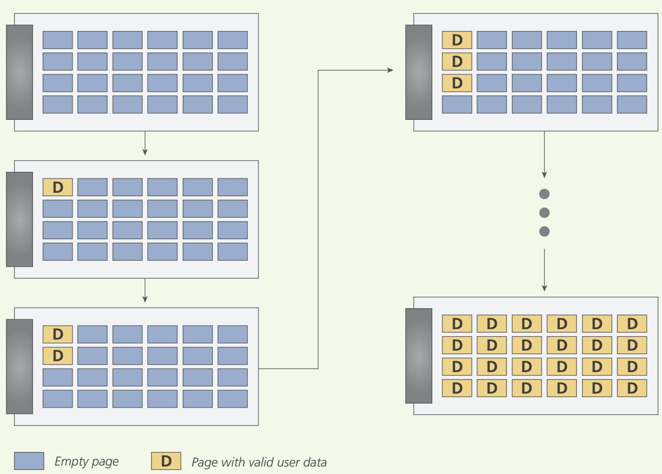


Figure 5: Filling a drive in random, small-block patterns.

Because the drive is empty, the host data is placed in the next-available empty page, eventually filling the drive with valid data whose logical block addresses (LBAs) are completely random (shown as yellow boxes with the letter D). In this diagram, a blue box is an empty page (ready to have data written), while a yellow box with a D is a page with valid data in it. At the end of this process, the drive is full.

Random data-deletion patterns

As the OS deletes data, the SSD marks it as invalid and hence as an erase candidate, shown as a green box with an I in the center. However, because of the random nature of the data, the pages marked as invalid follow no pattern or grouping and are selected completely at random.

Figure 6 shows an example of one such possible sequence—marking eight pages as invalid.

Because the traffic pattern in this example was random, the page-deletion pattern (in which pages are marked as invalid and hence as candidates for erasure) is also random and spread across the drive completely randomly. The result is a drive that looks rather like Swiss cheese and requires substantial garbage collection to prepare for block erasure to accept new, incoming data.

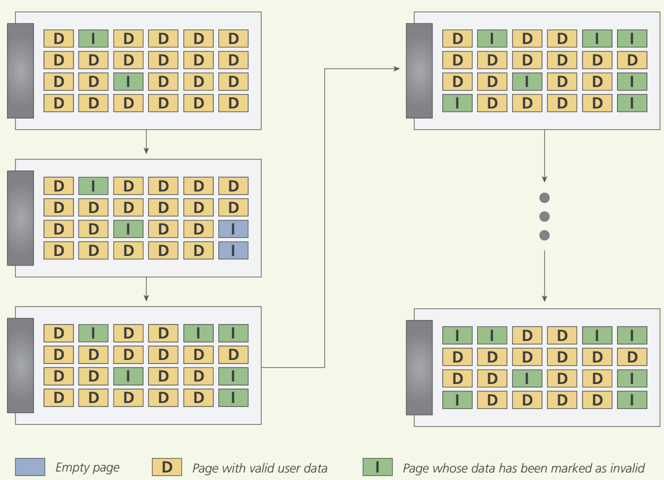


Figure 6: Marking pages as invalid.

Sequential patterns

In the example shown in Figure 7, an empty SSD is again filled one page at a time (with sequential data) until full.

As in the previous examples, blue boxes represent empty pages, while yellow boxes with a D in the center represent pages with valid data.

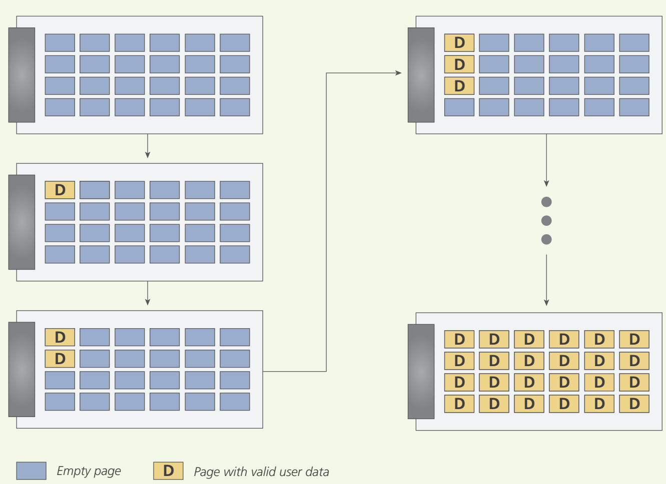


Figure 7: Filling a drive in sequential, small-block patterns.

Sequential data-deletion patterns

As the OS deletes data, the SSD marks it as invalid and hence as an erase candidate. In this case, because of the sequential nature of the data, the pages marked as invalid (green boxes with an I in the center) follow a distinct, grouped pattern and randomize the drive to a much smaller degree.

A natural consequence of writing data in orderly groups is that the data is also invalidated in an orderly manner (in this case, data is invalidated in sequential chunks—the same way it was written).

Because of this grouping, the effect of garbage collection to move valid data in preparation to receive new incoming data is substantially decreased, resulting in improved write performance.

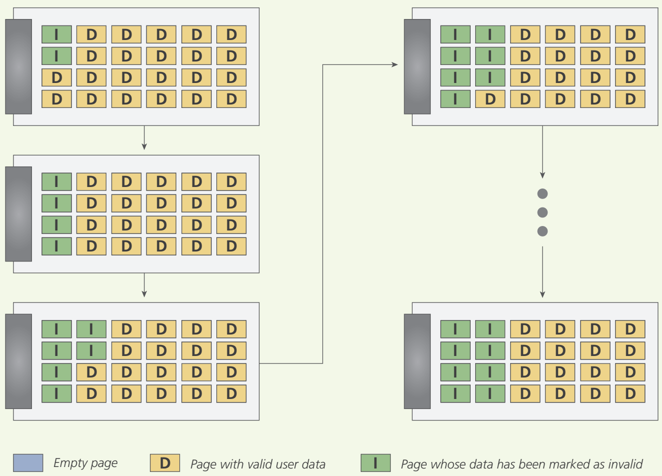


Figure 8: Orderly grouping resulting from a sequential data-deletion pattern.

Measured Write Performance and Host Patterns

The effects of host data patterns on write performance can easily be observed by constructing and examining write-saturation data for a given SSD. (For additional details on constructing write-saturation data, see the Micron technical marketing brief "Best Practices for SSD Performance Measurement.")

To illustrate this effect, two access patterns were analyzed, as shown in Table 1.

Random data-placement results: 4 KiB write

To evaluate the effect that the degree of write-traffic randomness has on SSD performance, the results of the two tests are shown below. The first test wrote the 4 KiB data sequentially, filling the drive (and rewriting it multiple times) in an orderly, well-grouped manner. The second test varied only the data placement—the same test sequence, transfer size, and I/O alignment were used, but the data was placed randomly, creating the Swiss cheese state mentioned previously.

SSD Tested	Transfer Size	Placement	Access
100GB enterprise SATA SSD (reference drive)	4 KiB (4096 bytes)	100% random	Write
		100% sequential	

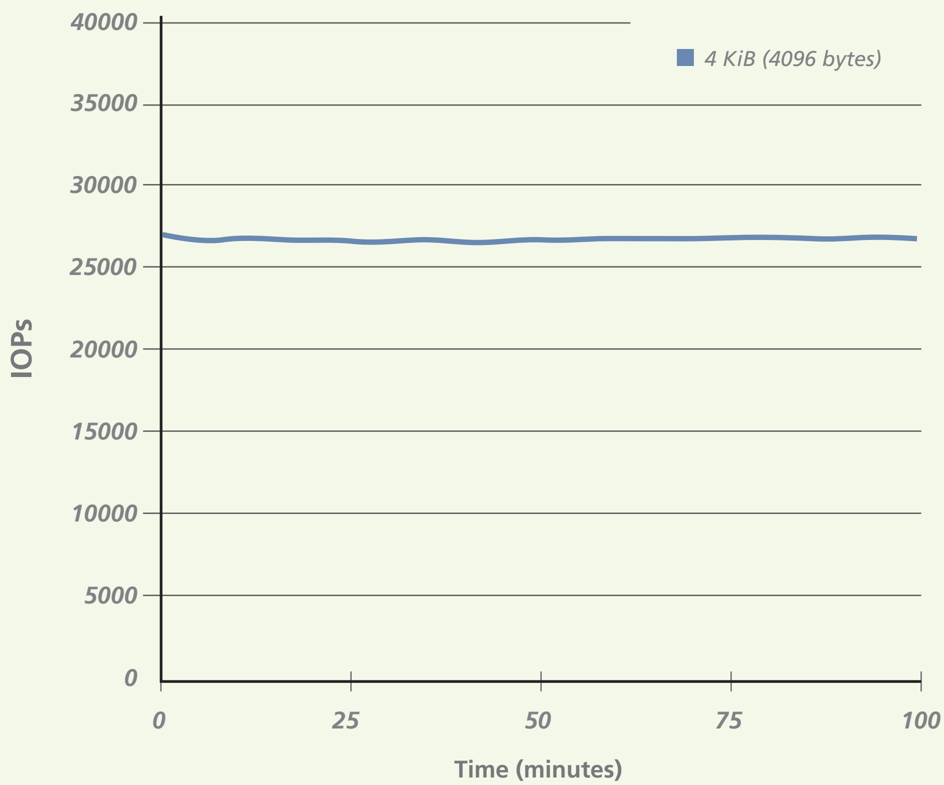
Table 1: Write access-pattern test parameters.

In each test, the write-saturation performance can be observed by plotting the stimulus results on a two-dimensional graph (the x-axis is in units of time; the y-axis is measures input/output operations per second, or IOPs).

Sequential data-placement results: 4 KiB sequential write IOPs

The first test wrote the data in well-ordered, sequential placement, substantially reducing the amount of work done by the garbage-collection process—because the data was placed sequentially, it was also invalidated sequentially. This write performance is shown in Figure 9.

Note: *The performance measured at the beginning of the test (about 26,000 IOPs) varies little from the performance measured after 100 minutes of sequential write traffic, also about 26,000 IOPs. This is a direct result of the orderly, sequential placement of the data and the minimal amount of work that has to be done by the garbage-collection process.*



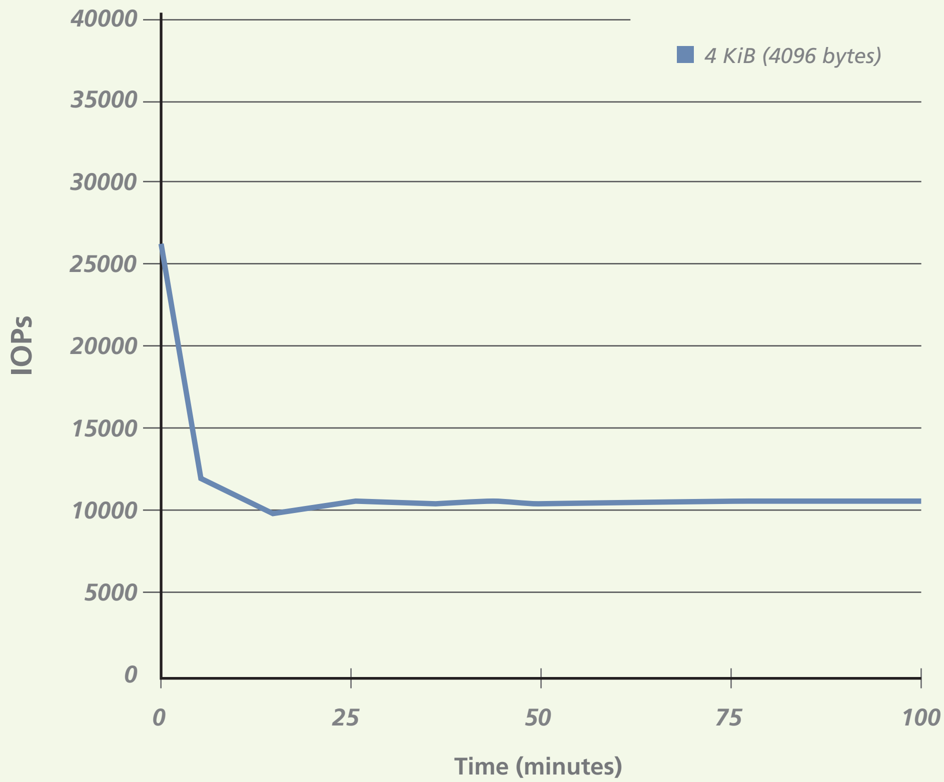
Enterprise SATA: Sequential Write

Figure 9: Improved write performance resulting from sequential placement of data.

Random data-placement results: 4 KiB random write IOPs

Now, if we use the same drive and apply the same stimulus and sequence, but instead place the data randomly, the steady-state, write-saturation performance is substantially different, as shown in Figure 10.

Note: *The substantial decrease in write IOPs as the drive is written, decreasing from an initial value of about 26,000 IOPs to a steady-state, write-saturation value of about 10,500 IOPs after 100 minutes of random write traffic. This is a direct result of the random placement of the data and the resulting random distribution of invalid data (when the host marks data as deleted).*



Enterprise SATA: Random Write

Figure 10: Decreased write performance resulting from random placement of data.

This dramatically increases the amount of work that has to be done by the garbage-collection process and decreases overall write performance.

Conclusion

Garbage collection and its effect on random write performance is an artifact of the write/erase disparity inherent in NAND-based SSDs. Because of the requirement that NAND with existing data in its cells (even data that the host has marked as invalid) must be erased before new data can be written, in addition to the requirement that NAND is erased in blocks rather than pages, a NAND-based SSD must periodically execute a process to group and migrate valid data (in pages) to a different location on the drive so that an entire block can be erased. This is the process of garbage collection.

When data is written randomly (and in small transfers), and the underlying data is then deleted, the associated pages are also invalidated (deleted) in the same random pattern. This results in a random scattering of valid/invalidated pages, forcing the SSD to collect a large number of small valid pages and move them to unoccupied areas of the drive so that the SSD can erase the block in preparation to write new, incoming data.

However, if data is written sequentially (even for common, small transfers of 4 KiB or 8 KiB), it is consequently invalidated in well-ordered groups, yielding a much more orderly distribution. This decreases the amount of work done by the garbage-collection process as it moves valid data to unoccupied areas of the drive. The reduction in the internal garbage-collection workload affords a marked increase in the SSD's write performance.

micron.com

12/12 EN.L P:12036