KIOXIA

Performance Brief

Decrease Write Amplification Factor / Improve Write Performance with KIOXIA XD8 Series SSDs and Flexible Data Placement

Introduction

When applications need data to be stored on an SSD, the host issues a write to a logical block which is mapped to a NAND block in the underlying storage device. If the data is moved or deleted by the host, that data becomes invalidated at the NAND block level, which in turn, can leave NAND blocks with a mix of valid and invalid data. Before a NAND block can be re-written, it must be fully erased. The process of moving valid host written data to another block, allowing that block to be erased to reclaim space occupied by invalid data, is referred to as garbage collection.

Performing garbage collection causes write amplification, where storing a single host write turns into many writes to the underlying flash media since the valid data must be written to a new location. The number of writes performed on the NAND media divided by the number of writes initiated by the host is a metric known as Write Amplification Factor (WAF). Higher values for WAF can cause unnecessary wear and reduce drive endurance, as well as degrade SSD performance and quality of service.

Flexible Data Placement (FDP) is a feature within the NVMe™ specification that allows a host system to provide hints to an SSD, such as how to group data, as opposed to the SSD making these choices on its own. By utilizing FDP, a host may be able to coordinate data placement with the SSD, and effectively lower WAF, depending on the workload. An ideal situation is for data to be organized so that entire superblocks (multiple flash blocks across several NAND die in an SSD) can become invalidated at the same time. This concurrent invalidation allows the flash blocks to be erased, without having to perform background garbage collection, since there is no remaining valid data to be copied to a new location.

By reducing the amount of garbage collection that occurs, WAF can ultimately be reduced, allowing an SSD to achieve longer device lifetime, improved performance through increased bandwidth available for host writes, reduced read/write latencies and reduced power consumption. FDP can more effectively segregate data from different applications which leads to a reduction of overall garbage collection.

This performance brief presents cumulative WAF test results using 1 DWPD[1] KIOXIA XD8 Series SSDs, with and without FDP enabled. To show the benefits that FDP can have on WAF, the KIOXIA Innovation Lab ran multiple tests demonstrating WAF changes when the Linux® kernel was compiled multiple times in a row, generating many temporary files. The tests also demonstrated the effect on WAF when swapping occurs due to imposed memory restrictions. An actual workload was used for the test: iteratively compiling the Linux kernel, which emulated a continuous integration/continuous deployment[2] environment. The test scenarios run included FDP enabled and disabled, with and without system memory restrictions.

Test Results Snapshot

KIOXIA XD8 Series SSDs delivered the following test results using Flexible Data Placement:

-

Cumulative WAF
(with memory restrictions)

~1.0
with FDP enabled

versus

2.8
with FDP disabled

Cumulative WAF
(without memory restrictions)

~1.0
with FDP enabled

versus

2.2
with FDP disabled

The test results show that the FDP-enabled KIOXIA XD8 Series SSD, with and without memory restrictions, was able to achieve a cumulative WAF that was close to 1, while the FDP-disabled KIOXIA XD8 Series SSD with memory restrictions had a cumulative WAF that was 2.8, and without memory restrictions, WAF was 2.2. In addition, FDP improves the performance of write-oriented workloads, so any workload that generates temporary files can benefit from FDP.

The test results also include a description of each test metric with graphical test results and analysis. Appendix A covers the hardware and software test configuration. Appendix B covers the configuration setup and test procedures.

Test Criteria

A few additional points regarding the tests:

- *Compilers often generate many temporary files, and typically, are no longer used post compilation. Writing these temporary files of various sizes mixed with files that have a longer lifespan can lead to increased WAF.*
- *Temporary files and swap spaces are used for data with short lifespans. With FDP, the host is able to hint the placement of these temporary files and swap spaces in the same media blocks. By separating this data from data with longer lifespans, WAF should be reduced.*

  - A swap space (i.e. swap memory or paging space) is a dedicated portion of computer storage that the OS can use to temporarily store inactive data from RAM when physical memory is full, and essentially acts as an extension of RAM, allowing the system to continue running smoothly even when RAM is exhausted.
- *One namespace was used for the OS root directory, and a second namespace was used to store the /tmp file system as well as the system swap space. For FDP-enabled drives, each namespace used a separate default Reclaim Unit Handle[3].*
- *The Linux® kernel is then compiled multiple times sequentially on the same test hardware with FDP disabled and enabled. A high level #include file was touched before each iteration to simulate edits and cause recompilation of large portions of the kernel.*
- *Tests were run, with and without memory restrictions, to show how WAF changes when just compiling the kernel and generating temporary files, as well as the additional effect on WAF when swapping occurs due to imposed memory restrictions.*

The four test scenarios included the following:

1. ***FDP Disabled without Memory Restrictions:***
   This scenario tests the KIOXIA XD8 Series SSD with FDP disabled and allows the host OS to use all available memory resources. When the workload compiles and recompiles the Linux kernel numerous times in a row, a current WAF baseline will be displayed, and the drive will not be able to take advantage of FDP.
2. ***FDP Enabled without Memory Restrictions:***
   This test scenario enables FDP on the KIOXIA XD8 Series SSD, with the workload re-run, to demonstrate that by simply enabling FDP can lower WAF significantly from the prior scenario.
3. ***FDP Disabled with Memory Restrictions:***
   This test scenario shows a WAF where memory is limited, the workload has temporary files being created and swap spaces are using the KIOXIA XD8 Series SSD to compensate for the lack of DRAM, which has a slightly higher WAF than when DRAM was abundant.
4. ***FDP Enabled with Memory Restrictions:***
   This test scenario shows that with FDP enabled, even in memory restricted environments, the KIOXIA XD8 Series SSD was able to be better utilized by the host OS, as swap spacing was also being used for temporary files, while still achieving a WAF that was close to 1 (the optimal goal of WAF). A WAF of 1 means that when the host issued a write, the KIOXIA XD8 Series SSD only had to perform 1 write worth of work and didn't have to do additional writes due to garbage collection.

The use of swap spaces for these test scenarios should be noted. In the first two scenarios, due to memory usage being unlimited (without memory restrictions), the use of swap spaces is not evident, just the effects of FDP when it comes to the creation of temporary files during the Linux kernel compilation. For these test scenarios, both temporary files and swap spaces were placed on an available namespace. The Linux kernel compilation process used many temporary files to complete the compilation, and the first two scenarios demonstrate that effect on WAF.

The last two test scenarios demonstrate how WAF is impacted in test environments where available memory is low (with memory restrictions), as low memory availability causes the OS to swap frequently from DRAM to the KIOXIA XD8 Series SSD.

Three metrics were recorded for each of the four test scenarios as follows:

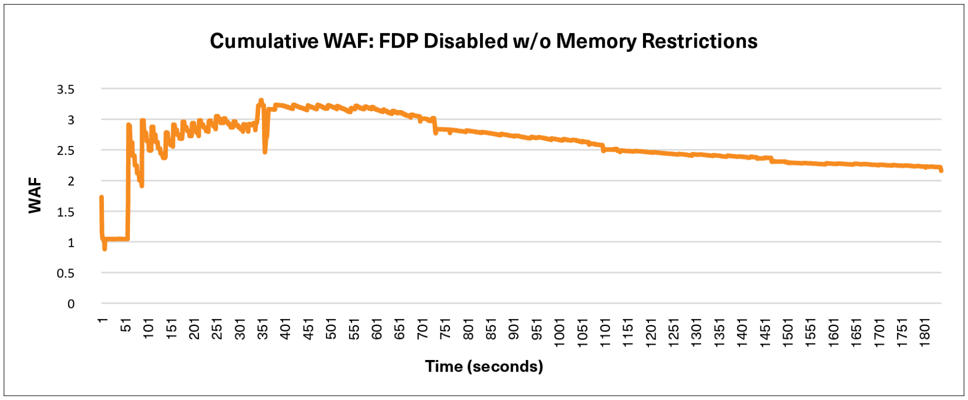
1. ***Cumulative WAF:***
   This metric refers to the media writes and host-issued writes that occurred from the start of the workload. Cumulative WAF is important as it shows the WAF that is generated by the workload, which can be used to see how many more media writes are occurring relative to host-initiated writes. WAF values equal or close to 1 indicate that the number of media writes and host-initiated writes are equal or close to equal of each other, while a higher WAF value indicates more media writes are occurring than host-initiated writes. When this occurs, both the lifetime of the underlying flash memory and drive endurance are affected.

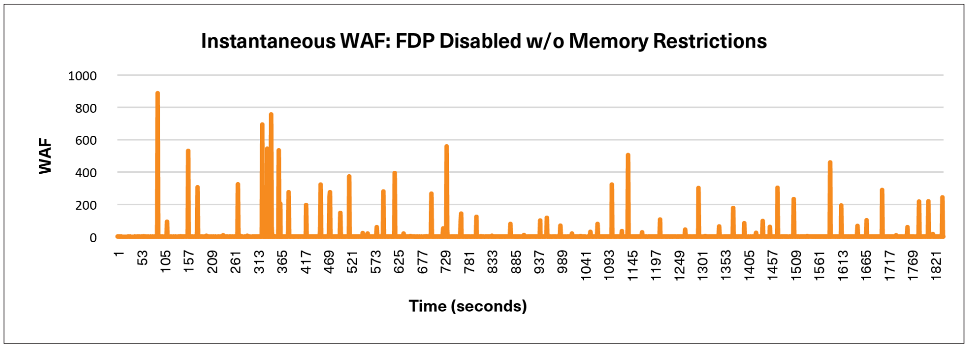
2. ***Instantaneous WAF:***
This metric refers to the media writes and host-issued writes that occurred from the last time WAF was measured. It provides a look inside garbage collection trends for a given workload. Portions when instantaneous WAF are high indicate that many more media writes were issued by the device itself, demonstrating increased garbage collection.
3. ***Clamped Instantaneous WAF:***
As some of the instantaneous WAF values were very high, which can impact scaling, these values were clamped to a maximum value of 5 to better visualize the variance of WAF over time for the given workload.

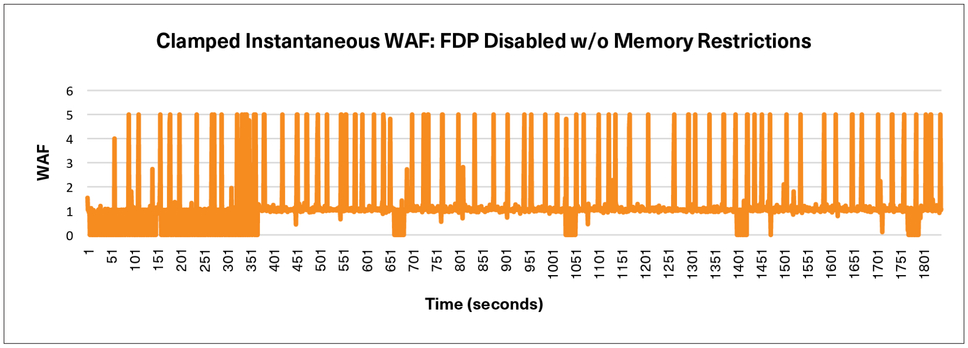
Test Results[4] / Analysis

FDP Disabled without Memory Restrictions

When FDP was disabled and had no memory restrictions, the cumulative WAF sat at about 2.2, demonstrating that during the kernel compilation, the media was being written to much more than just what the host was requesting to write, mainly due to the creation of many temporary files that triggered garbage collection. The instantaneous WAF showed very high peaks throughout the workload, highlighting the dramatic increases in media writes as compared to host writes for the given workload. Clamping the results, where the scaling on the y-axis was set to a maximum of 5, showed that there were frequent peaks in WAF, demonstrating a trend of frequent garbage collection that occurred.



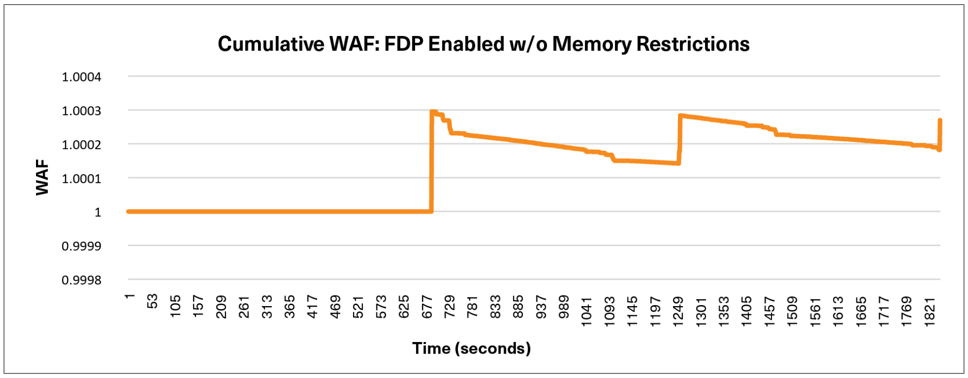


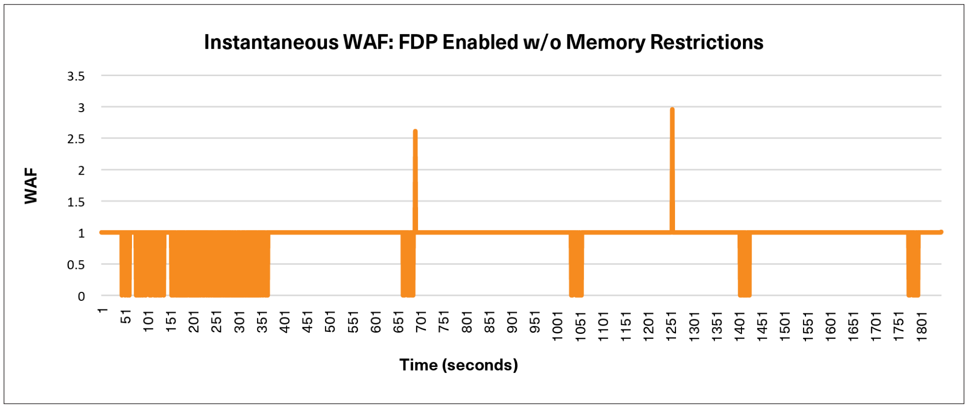


FDP Enabled without Memory Restrictions

When FDP was enabled and the system had no memory restrictions, WAF was dramatically decreased versus FDP disabled. As seen from the graph below, the cumulative WAF sat at ~1.0, demonstrating that during the kernel compilations, FDP was able decrease the WAF by a considerable amount.

It should be noted that this test scenario did not include a clamped instantaneous WAF test result. Clamped instantaneous WAF was only included if the measurement interval happened to have a measurement that was incredibly high, when compared to the rest of the values. The clamped instantaneous WAF of the other three scenarios depicted more of a trend in the data with respect to the workload and the present state of WAF.





Many periods of zero writes were also observed that occurred due to the Linux® OS having a very effective bcache, as all unused DRAM was utilized by this capability. Bcache, or block cache, is a Linux kernel feature that typically enables faster access to commonly used data by keeping a copy in DRAM rather than accessing it from storage devices. It's designed to improve the read/write performance and may extend the life of flash-based devices. For the tests that had no memory restrictions, there were periods where no write I/O was issued to the KIOXIA XD8 Series SSD because the writes were being cached in DRAM. The bcache approach for these tests was very effective because the test system had an abundance of DRAM to use.

The first test scenario, FDP Disabled without Memory Restrictions, cached a lot of writes, and by the end of each build, the workload became more read dominated. Swapping did not occur as often due to the abundance of DRAM available. For the second test scenario, FDP Enabled without Memory Restrictions, WAF got very close to 1, showing the effectiveness that FDP had on this workload.

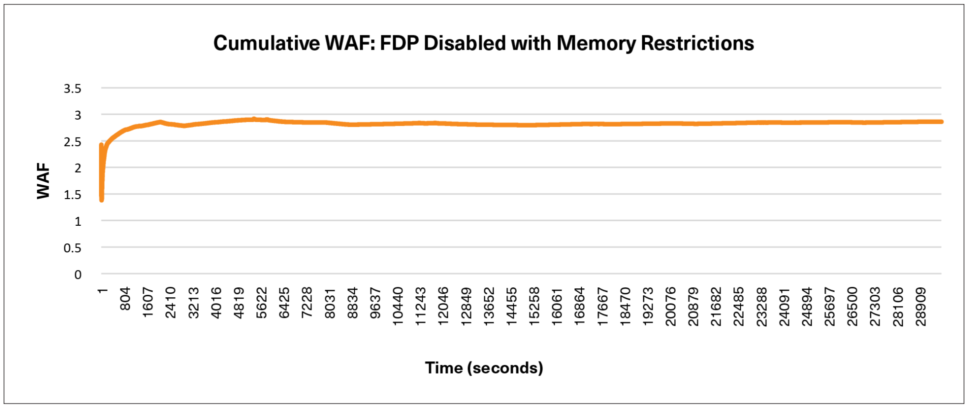
To incur swapping more often, memory restrictions were imposed to see the result it would have on WAF.

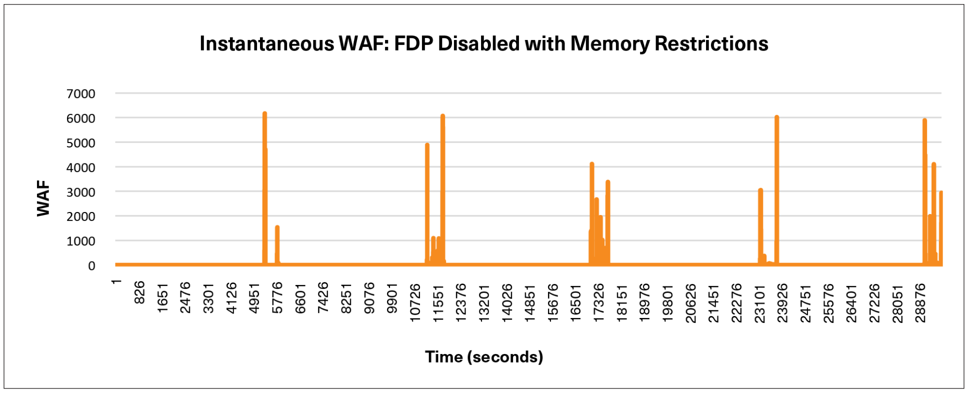
FDP Disabled with Memory Restrictions

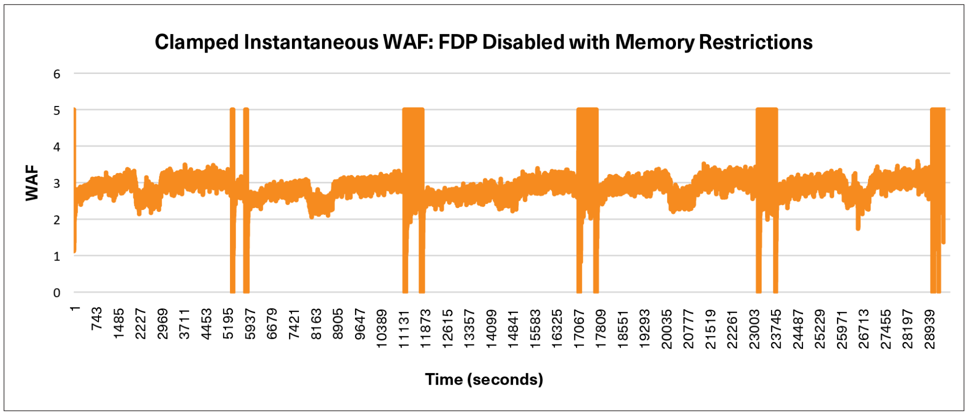
For the third and fourth test scenarios with memory restrictions, DRAM usage was limited to 8 GB to see how swap space utilization impacted WAF results.

For this test scenario, the cumulative WAF sat at 2.8, showing that WAF got even higher once bcache did not have an abundance of DRAM to use. Swapping also introduced dependencies between I/O operations as a swap-in cannot occur until the swap-out has completed. The result caused increased latencies due to garbage collection, which in turn blocked processes and reduced CPU utilization.

Instantaneous WAF for this test scenario was very large, showing very high maximums. Dips below a WAF of 1 on the graph indicated that there were no host writes during that specific sample interval. These high maximums made it hard to see the underlying data, so a clamped instantaneous WAF test was used to zoom in to the results and better observe the drive behavior. For this clamped result, the WAF was almost always 3, and the peaks evident in the graph showed frequent garbage collection that occurred. Frequent garbage collection will wear the underlying media much quicker and is also the cause for major performance degradation.

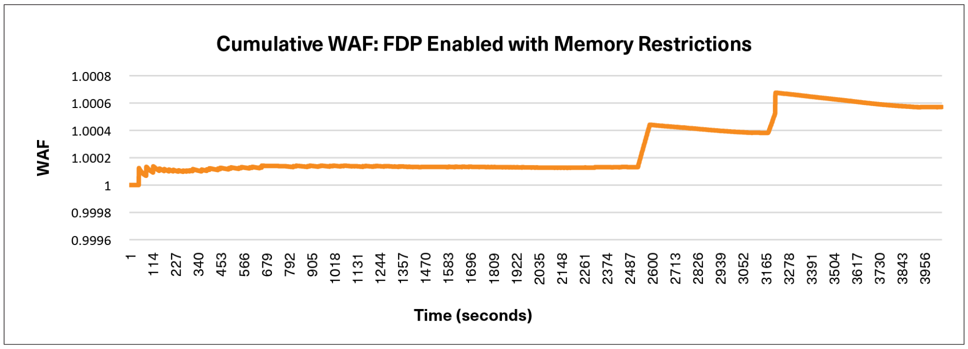


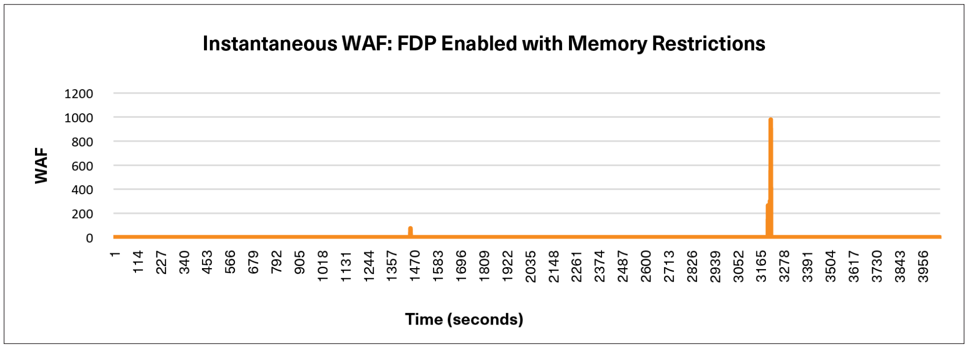


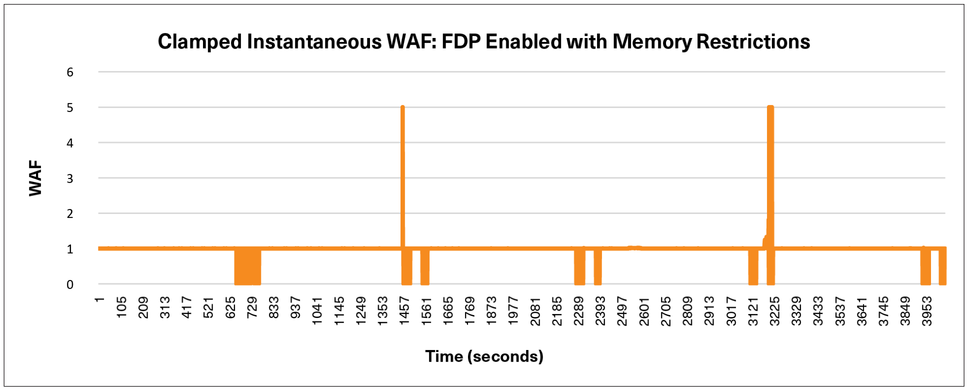


FDP Enabled with Memory Restrictions

By comparison, for this test scenario, WAF comfortably sat close to 1. The previous workload run with FDP disabled generated a WAF of 2.8, which showed that enabling FDP for this workload can effectively more than double the lifespan of the underlying KIOXIA XD8 Series SSD. Instantaneous WAF also sat very comfortably most of the time at a WAF close to 1, as very little garbage collection occurred and evident from the lack of spikes or peaks in the graph.







The four test scenarios yielded these results:

With Memory Restrictions	Cumulative WAF	Without Memory Restrictions	Cumulative WAF
FDP Disabled	2.8	FDP Disabled	2.2
FDP Enabled	~1.0	FDP Enabled	~1.0

By lowering the cumulative WAF from 2.2 to ~1.0 when there were no system memory restrictions, the 1 DWPD KIOXIA XD8 Series SSD with FDP being enabled must do approximately half of the writes it would have done for the given workload if FDP was disabled. Similarly, the system with memory restrictions was able to lower the cumulative WAF from 2.8 to ~1.0 and only had to do one-third of the writes with FDP being enabled for the given workload. In these scenarios, reducing the number of writes to the SSD effectively will reduce the amount of garbage collection and increase drive endurance, allowing for longer lifetimes of individual SSDs.

Summary

KIOXIA XD8 Series SSDs with FDP enabled effectively decreased WAF versus when FDP was disabled. The additional benefit of FDP is improved performance through increased bandwidth that it makes available for host writes so any workload that generates temporary files can benefit from FDP.

KIOXIA XD8 Series SSD Product Information[5]

The third generation KIOXIA XD8 Series SSDs are PCIe® 5.0 and NVMe™ 2.0 specification compliant and are designed to the Enterprise and Datacenter Standard Form Factor (EDSFF) E1.S specification. This enables KIOXIA XD8 Series SSDs to address specific requirements of hyperscale applications. These SSDs are available in a 9.5 mm heatsink option and a 15 mm heatsink option. A 25 mm heatsink option will be available to select customers. The series features up to 7.68 terabyte[6] (TB) capacities, sequential read performance up to 12,500 megabytes[6] per second (MB/s), sequential write performance up to 5,800 MB/s, random read performance up to 2,300,000 IOPS, random write performance up to 250,000 IOPS and 1 DWPD of endurance. KIOXIA XD8 Series SSDs represent the future of flash storage for servers and storage systems in cloud and hyperscale data centers. More product information is available [here]() and [here]().

KIOXIA XD8 E1.S Series SSDs[7]
9.5 mm (left) / 15 mm (right)

Appendix A

Hardware/Software Test Configuration

Server Information	
Model	Supermicro® SSG-121E-NES24R
No. of Servers	1
No. of CPU Sockets	2
CPU	Intel® Xeon® Gold 6444Y
No. of CPU Cores	16
CPU Frequency	3.6 GHz
Total Memory	128 GB[6] DDR5 DRAM
Memory Frequency	DDR5-4800

Operating System Information	
Operating System	Ubuntu®
Version	24.01 LTS
Memory Limit	mem=8G*
Swap Size	1M * 262,144 GB
libnvme version	1.11.1
nvme-cli version	2.11
Compiled Kernel version	6.12.1
make version	4.3
nproc used for make	64

SSD Information	
Model	KIOXIA XD8 Series
Interface	PCIe® 5.0 x4
Protocol	NVMe™ 2.0
No. of Drives	2 *(1 for FDP enabled / 1 for FDP disabled)*
Form Factor	EDSFF E1.S
Capacity	3.84 TB[6]
DWPD	1 (5 years)
Power Consumption	20 W
No. of Namespaces	2
Namespace #1 Size	100 GB
Namespace #2 Size	3,740 GB

**For constrained use case only.*

Appendix B

Configuration Setup/Test Procedures

Configuration Setup

Two KIOXIA XD8 Series SSDs were installed into a Supermicro® SSG-121E-NES24R server.

- One drive was set up with FDP disabled
- The other drive was set up with FDP enabled
- Both SSDs had two namespaces created, where one was used to host the OS and the other was used to host tmp files and swap spaces.

Both SSDs were set up with an Ubuntu® 24.01 LTS operating system.

Each KIOXIA XD8 Series SSD was pre-conditioned.

The source code for Linux® compiled kernel version 6.12.1 was downloaded locally and the kernel compilation was started.

Test Procedures

The four test scenarios were performed for both KIOXIA XD8 Series SSDs, with FDP disabled and enabled, as well as with memory restrictions and without.

The metrics collected included cumulative WAF and instantaneous WAF.

NOTES:

[1] DWPD: Drive Write(s) Per Day. One full drive write per day means the drive can be written and re-written to full capacity once a day, every day, for the specified lifetime. Actual results may vary due to system configuration, usage, and other factors.

[2] The workload used to compile the Linux® kernel was an attempt to emulate a continuous integration/continuous deployment environment, which features a set of practices for automating the software development process and make it faster and more reliable. Whenever the Linux kernel is compiled, integration and test steps of these practices are conducted, and what the workload aimed to emulate.

[3] A Reclaim Unit (RU) can be equal to SSD superblocks and the Reclaim Unit Handle acts as pointers for the RU allowing the host to write to multiple RUs at a time.

[4] Read and write speed may vary depending on the host device, read and write conditions and file size.

[5] KIOXIA XD8 Series SSD performance specifications are publicly available and accurate as of this publication date. Specifications are subject to change. Read and write speed may vary depending on the host device, read and write conditions, and file size.

[6] Definition of capacity: Kioxia Corporation defines a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes, a terabyte (TB) as 1,000,000,000,000 bytes and a petabyte (PB) as 1,000,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1Gbit = 2^{30} bits = 1,073,741,824 bits, 1GB = 2^{30} bytes = 1,073,741,824 bytes, 1TB = 2^{40} bytes = 1,099,511,627,776 bytes and 1PB = 2^{40} bytes = 1,125,899,906,842,624 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

[7] The product image shown is a representation of the design model and not an accurate product depiction.