Course Project #3: SSD Performance Profiling
21 October 2024

1 Introduction

The goal of this project was to profile the performance of my SSD by measuring its latency and bandwidth under different conditions. Specifically, I wanted to understand how data access size, read/write intensity ratio, and I/O queue depth affect the SSD's performance.

Initially, I planned to use the Flexible IO Tester (FIO) as per the project guidelines. However, I encountered compatibility issues while attempting to install and run FIO on my MacBook running macOS Monterey. FIO relies on certain Linux-specific kernel features and system calls that are not available on macOS, such as `libaio` and `io_uring` interfaces. These dependencies are essential for FIO's advanced I/O testing capabilities. Despite exploring various installation methods, including using package managers like Homebrew and attempting to compile from source, I was unable to get FIO operational on my system. The lack of native support for Linux asynchronous I/O interfaces in macOS was a significant barrier. Additionally, macOS's security features and differences in filesystem architecture posed further challenges. To overcome this limitation, I decided to develop custom Python scripts to simulate the I/O operations and collect performance data. Python provides cross-platform support and allows for precise control over file I/O operations, making it a suitable alternative for this project.

To make the experiments manageable, I chose to vary one parameter at a time while keeping the others fixed. This approach allowed me to isolate the effect of each parameter on performance without running all possible combinations (which would be $4^3 = 64$ experiments). By fixing certain parameters, I could focus on the specific impact of the variable under study, ensuring that the latency measurements were dominated by the intended factor.

2 Experiment 1: Effect of Data Access Size on Latency and Bandwidth

2.1 Experimental Setup

In this experiment, I investigated how changing the data access size affects latency and bandwidth. I tested three different access sizes: 4KB, 16KB, and 128KB. I fixed the read/write ratio at 100% reads (read-only) because I wanted to eliminate any variability introduced by write operations. Reads are generally more consistent and have less overhead compared to writes. Additionally, I kept the I/O queue depth minimal by performing single-threaded operations. This was done to ensure that the latency measurements were dominated by the actual I/O operation time rather than waiting in a queue. By fixing these parameters, I aimed to focus solely on the impact of data access size.

2.2 Results

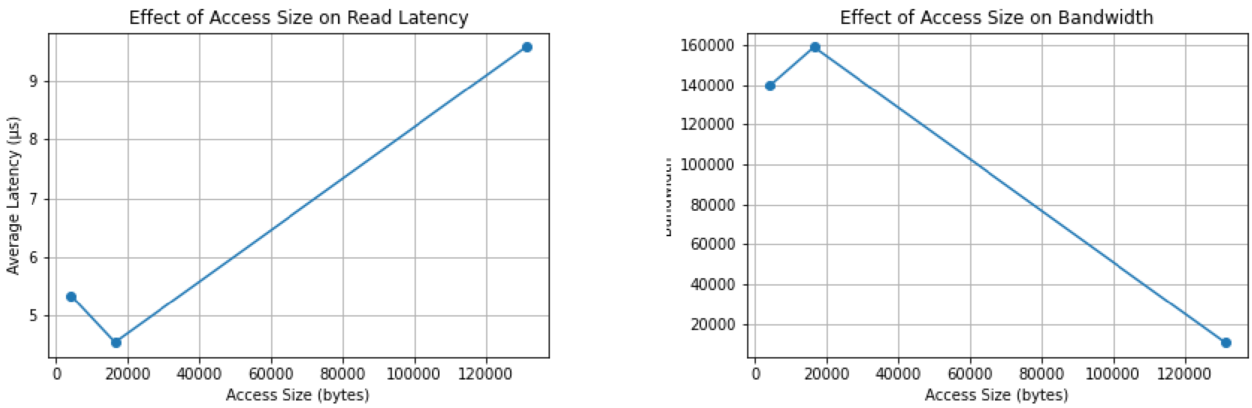
The average latency and bandwidth measured for each access size are shown in Table 1.

Access Size (bytes)	Average Latency (μs)	Bandwidth (IOPS)
4,096	5.35	139,592.99
16,384	4.55	158,848.56
131,072	9.57	10,865.94

Table 1: Effect of Data Access Size on Latency and Bandwidth

2.3 Analysis

From the results (see Figures 1a and 1b), I observed that the average latency decreased slightly when increasing the access size from 4KB to 16KB, but then increased when the access size was increased to 128KB. This suggests that there's an optimal access size where latency is minimized. The initial decrease in latency might be due to more efficient use of the SSD's internal mechanisms when handling slightly larger blocks of data. However, when the access size becomes too large, the time taken to transfer the data dominates, leading to increased latency.



(a) Average Latency (b) Bandwidth (IOPS)

Figure 1: Effect of Access Size on Performance Metrics

For bandwidth, measured in IOPS, the IOPS increased when increasing the access size from 4KB to 16KB, indicating that the SSD can handle more operations per second with larger access sizes in this range. However, at 128KB, the IOPS decreased significantly. This is expected because larger access sizes result in fewer I/O operations per second. Although each operation transfers more data, the number of operations that can be completed per second decreases. The decrease in IOPS at 128KB reflects this trade-off between operation size and operation count.

3 Experiment 2: Effect of Read/Write Intensity Ratio on Latency and Bandwidth

3.1 Experimental Setup

This experiment aimed to understand how different read/write ratios affect SSD performance. I tested four ratios: 100% reads, 70% reads / 30% writes, 50% reads / 50% writes, and 100% writes. I fixed the data access size at 4KB because small access sizes are common in real-world applications and would make the latency differences more noticeable. The I/O queue depth was kept minimal to ensure that the latency measurements reflected the operation time rather than queuing delays. By fixing these parameters, I focused on how the mix of read and write operations impacts performance.

3.2 Results

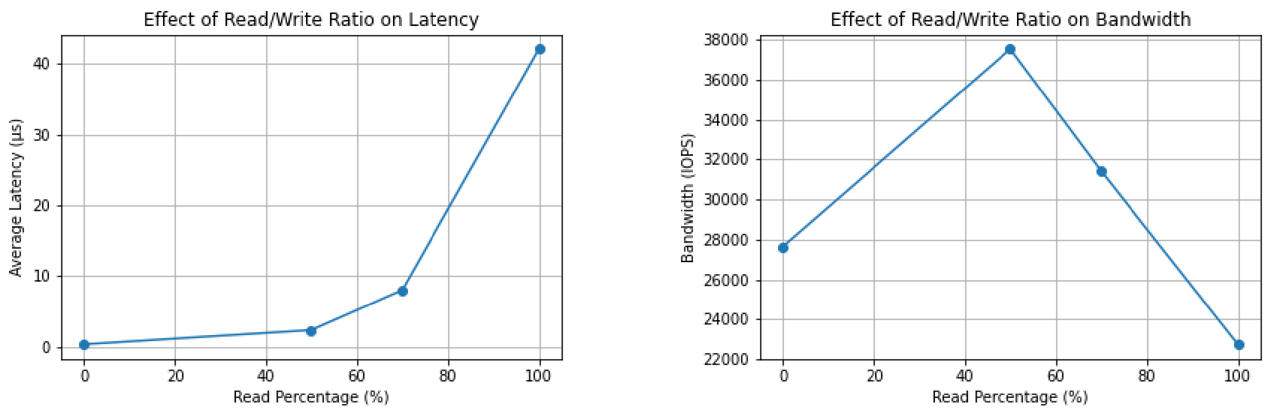
The average latency and bandwidth for each read/write ratio are presented in Table 2.

Read Ratio (%)	**Average Latency (μs)**	**Bandwidth (IOPS)**
100	42.05	22,735.91
70	7.99	31,419.10
50	2.33	37,541.60
0	0.31	27,638.66

Table 2: Effect of Read/Write Ratio on Latency and Bandwidth

3.3 Analysis

The results (see Figures 2a and 2b) show that as the percentage of reads decreases, the average latency decreases significantly. At 100% reads, the latency is highest at 42.05 μs, while at 100% writes, it drops to 0.31 μs. This suggests that in my SSD, write operations are faster than read operations, which might be due to write caching mechanisms that make writes appear faster.



(a) Average Latency (b) Bandwidth (IOPS)

Figure 2: Effect of Read/Write Ratio on Performance Metrics

Bandwidth, measured in IOPS, increases as the read/write ratio becomes more balanced, peaking at 50% reads and 50% writes with 37,541.60 IOPS. This indicates that the SSD performs best with a mix of read and write operations, possibly because it can optimize internal processes when handling both types of requests. At 100% writes, the bandwidth decreases compared to the peak, possibly due to write amplification or internal write management overhead.

4 Experiment 3: Effect of I/O Queue Depth on Latency and Bandwidth

4.1 Experimental Setup

In this experiment, I explored how varying the I/O queue depth affects performance. I tested queue depths of 1, 10, and 100. I fixed the data access size at 4KB and the read/write ratio at 100% reads. I chose 100% reads to eliminate any variability introduced by write operations and to focus solely on how queue depth impacts read performance. By keeping the access size small, I ensured that the latency measurements were sensitive to changes in queue depth.

4.2 Results

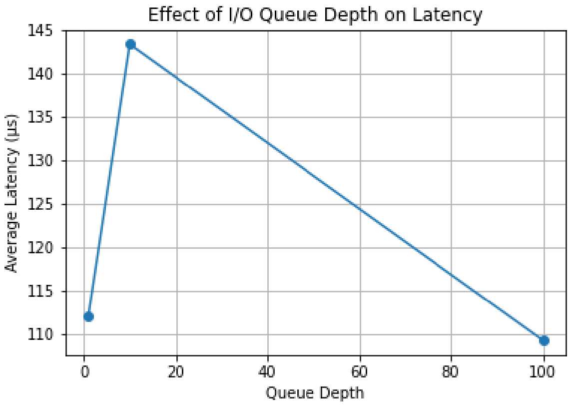
The average latency and bandwidth for each queue depth are shown in Table 3.

Queue Depth	Average Latency (μs)	Bandwidth (IOPS)
1	111.93	5,216.89
10	143.36	11,383.80
100	109.22	11,944.40

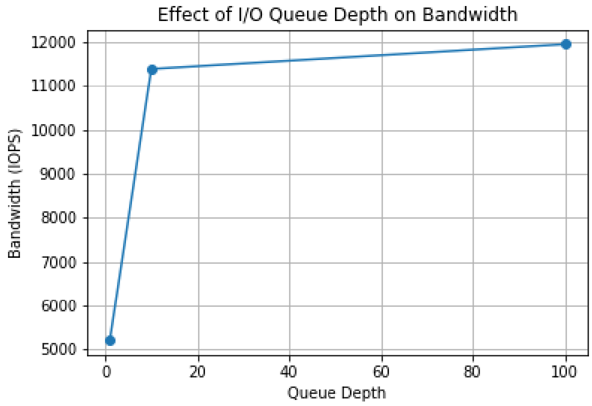
Table 3: Effect of I/O Queue Depth on Latency and Bandwidth

4.3 Analysis

The results (see Figures 3a and 3b) indicate that increasing the queue depth from 1 to 10 leads to an increase in both latency and bandwidth. The average latency increases from 111.93 μs to 143.36 μs, likely due to more requests waiting in the queue and increased contention for resources. However, the bandwidth more than doubles, showing that higher queue depths allow the SSD to process more operations in parallel, improving overall throughput. When the queue depth is increased to 100, the average latency decreases slightly to 109.22 μs, and the bandwidth increases marginally. This decrease in latency at a higher queue depth may be due to the SSD's ability to better optimize and schedule a large number of outstanding requests, effectively



(a) Average Latency



(b) Bandwidth (IOPS)

Figure 3: Effect of I/O Queue Depth on Performance Metrics

reducing the time each request spends waiting. It suggests that there is an optimal queue depth where the SSD's internal parallelism is fully utilized.

5 Comparison with Intel Data Center NVMe SSD D7-P5600

The Intel Data Center NVMe SSD D7-P5600 specifies a random write-only 4KB IOPS of 130,000. In my experiments, the highest IOPS I measured was around 37,541.60 IOPS at a 50% read/write ratio. This is lower than the Intel SSD, which is expected because the Intel SSD is an enterprise-grade drive designed for higher performance and reliability. My SSD is a consumer-grade drive, which typically has lower performance characteristics due to differences in technology, caching mechanisms, and firmware optimizations.

6 Conclusion

Through these experiments, I learned how different factors affect SSD performance. Changing the data access size can optimize latency and bandwidth up to a point, but too large access sizes can increase latency due to data transfer time and reduce IOPS. The read/write ratio significantly impacts performance, with write operations being faster in my SSD, possibly due to caching mechanisms, and a balanced mix of reads and writes providing the highest IOPS. Lastly, increasing the I/O queue depth can improve bandwidth but may also increase latency due to queuing delays, although at higher depths, the SSD might optimize request handling to reduce latency. Understanding these performance characteristics is important for optimizing systems based on specific workloads. By knowing how these parameters affect performance, we can make informed decisions when configuring storage systems.