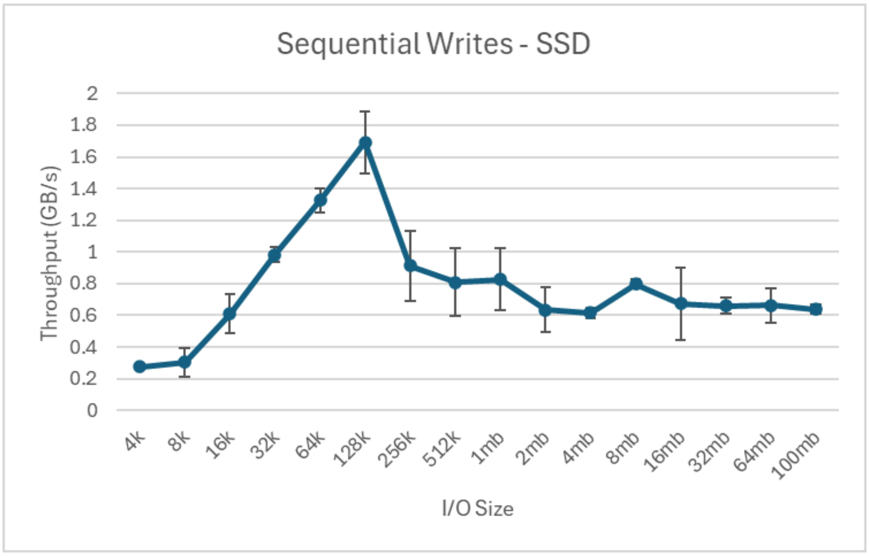
COMP 530 Lab #5 - SSD Speed Test Writeup

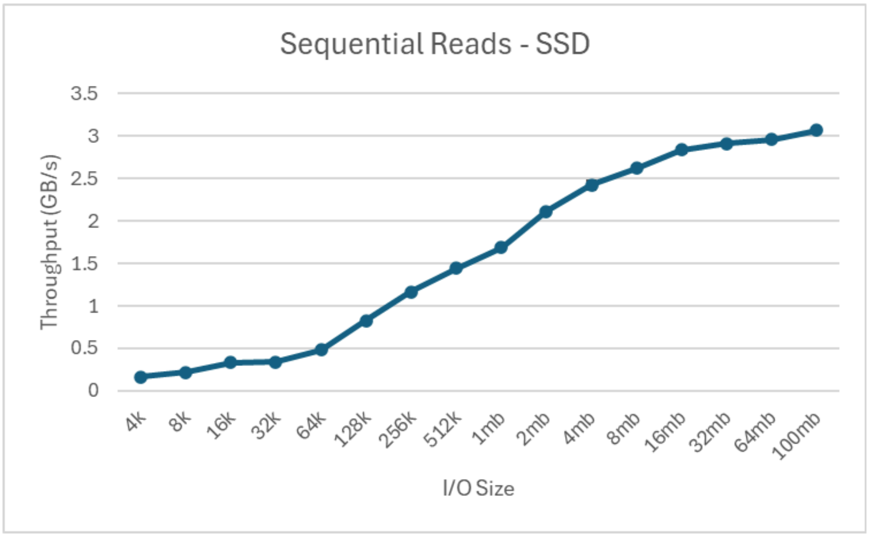
PIDs: 730481231, 730466078

For our tests, we used an M1 Macbook Air with Apple's proprietary 256 GB SSD with purported sequential write and read speeds of 2910 MB/s and 2676 MB/s, respectively. We ran tests for sequential, stride, and random writes and reads to the SSD.

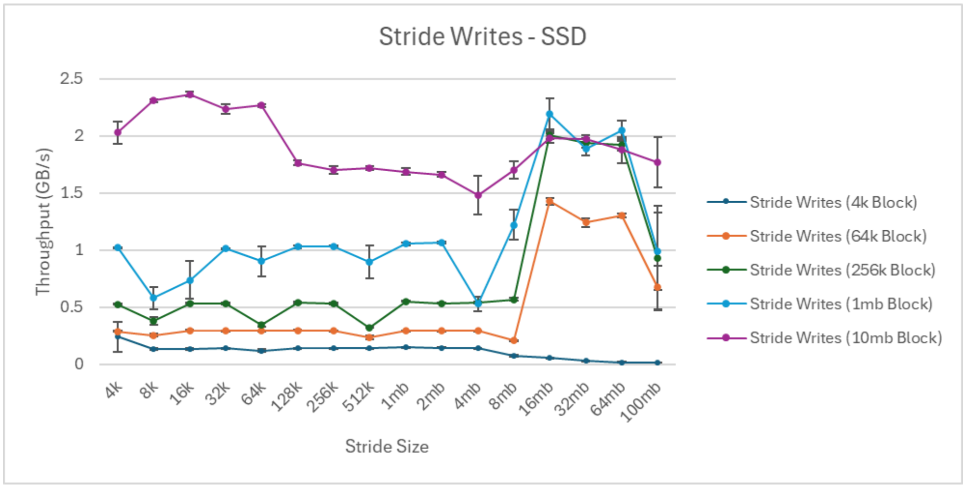


For our sequential writes test, we gathered ten samples for each data point on the chart. Based on this test, it seems like the optimal I/O block size for a series of sequential writes is 128 kilobytes. The throughput steadily increases—almost linearly—from 4 kilobytes to this point, then drops off as sizes get larger.

SSDs can be optimized to write certain data sizes, so it's possible that the M1 Macbook Air's SSD is optimized to write 128 kilobyte blocks of data. This could be why the throughput peaks at 128 kilobytes and is lower at smaller and larger sizes.



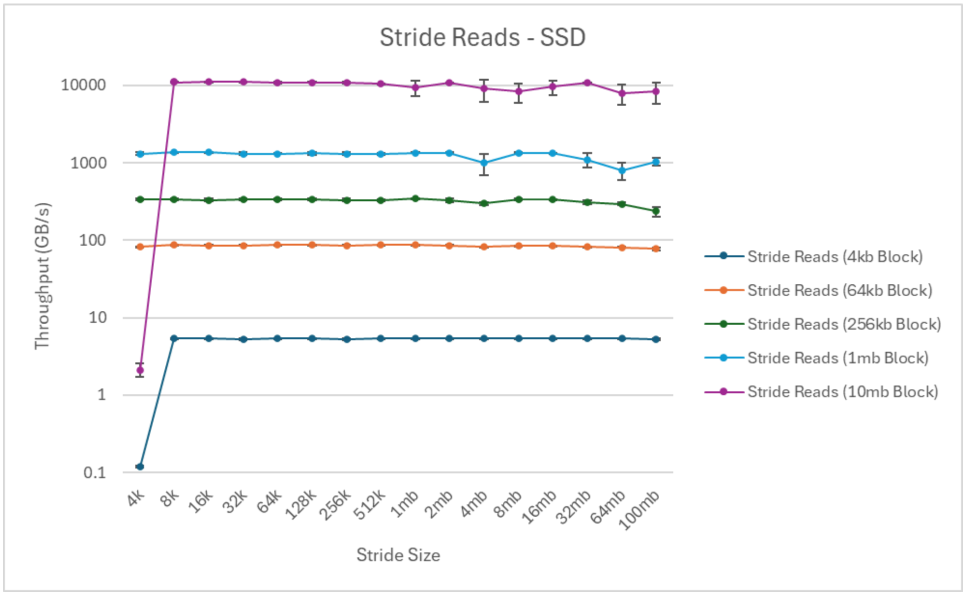
For our sequential reads test, we gathered ten samples for each data point on the chart. The throughput for reads steadily increased all the way from 4 kilobyte reads to 100 megabyte reads. It stands to reason that this is the case because reading large blocks of memory requires less overhead from the SSD, allowing throughput to increase.



For our stride writes test, we gathered five samples for each data point. Generally, as the stride size increased, the throughput tended to increase, apart from data for the largest writes of

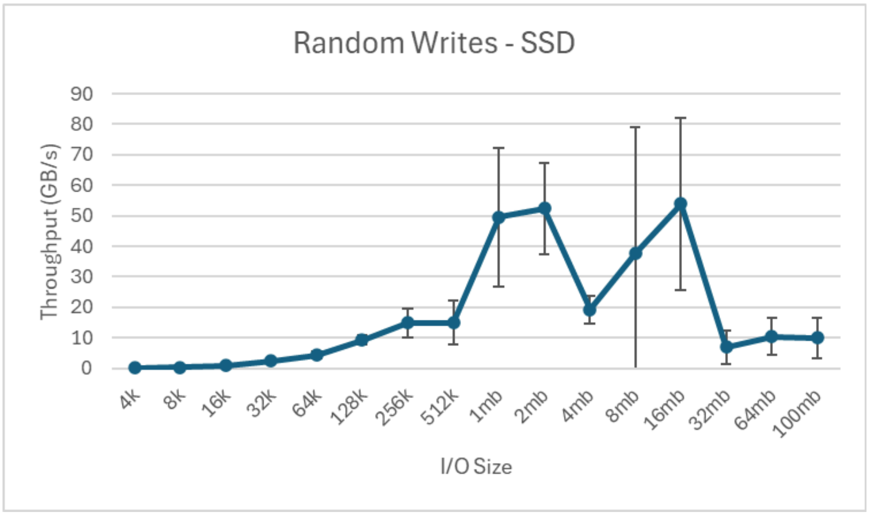
10 megabytes and the smallest writes of 4 kilobytes. Since SSDs generally perform better when writing larger blocks to memory, the trend for 10 megabyte writes is unexpected.

Interestingly, for the four larger block sizes, the throughput peaked when the stride was 16 megabytes. It is possible that this aligns well with the optimal write size of the SSD, or that a combination of the stride and write size is aligned with the SSD.

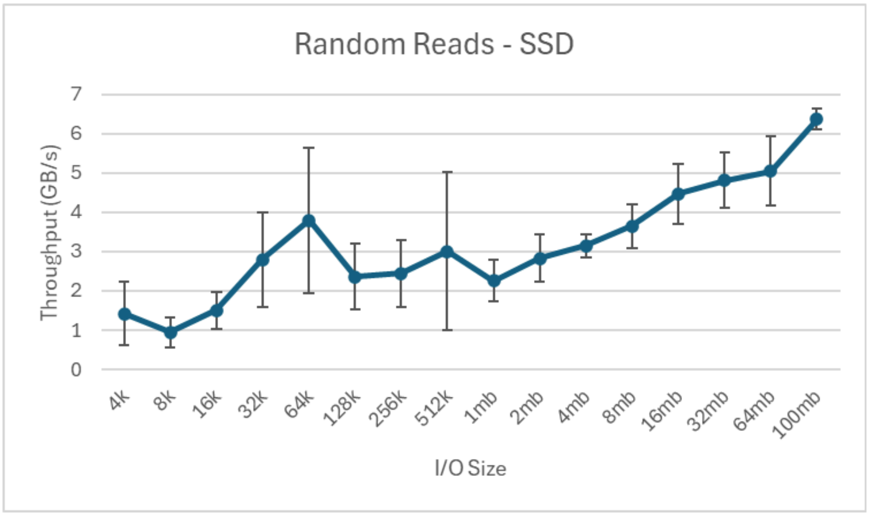


For our stride reads test, we gathered five samples for each data point. The y-axis of the graph is logarithmic in scale to accommodate the range of throughput.

The read speed of the SSD seems to remain fairly constant across stride size, barring the performance of 4 kilobyte strides for the smallest and largest reads (of 4 kilobytes and 10 megabytes, respectively). Compared to the sequential reads, this data makes sense. Read throughput increases as the size of the read increases, but seeking across a file with stride reads interrupts the workflow and causes the curve to flatten out.



For our random writes test, we gathered ten samples for each data point on the chart. The throughput increases steadily from 4 kilobyte writes up until 256 kilobytes, then varies. Writing to random locations on an SSD increases latency and is slower than sequential or stride writes, so random writes should have lower throughput than our previous write tests. Except for data from the lowest-sized writes, the data we have does not reflect this, so it is possible that we somehow wrote to cache instead of disk for this test.



For our random reads test, we gathered ten samples for each data point on the chart. As with the sequential reads, the throughput steadily increases as the read sizes increase. In our data, the random reads reach a higher throughput than the sequential reads, which should not happen. Again, like with our random writes test, it is possible that we are reading from cache instead of the SSD somehow.

In conclusion, it seems that sequential reads and writes are likely the most optimal. The read test especially showed expected data, as throughput increased with read size due to the decreasing presence of overhead. The sequential write test showed that the M1 Macbook Air's SSD could potentially be optimized for 128 kilobyte writes. The read test for stride reads also displayed expected values. However, our random tests might have been affected by caching, as they resulted in unexpectedly fast throughput. Overall, our tests showed that different access patterns (and potentially caching) have different effects on throughput.