Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

Project 03

Introduction:

There is a clear tradeoff between bandwidth and latency when utilizing memory or storage. Using queueing theory, it can be determined that as the bandwidth increases, the latency for each request increases significantly.

Two programs that can be used to test the bandwidth and latency of memory and storage are Intel Memory Latency Checker and Flexible IO Tester (FIO). The Memory Latency Checker can be used to experimentally determine the bandwidth and latency of the memory, and FIO can be used to determine those properties of storage.

Hardware Environment:

CPU:	Intel Core i9-9880H @ 2.30GHz
Memory:	31.9GB DDR4
Storage:	256GB SSD
Cache:	L3 – 16MB
	L2 – 2MB
	L1 – 512KB

Memory tests performed on Windows Boot Camp.
Storage tests performed on Linux with a 20GB segment of the SSD.

Settings:

Intel Memory Latency Checker

- -Tested at 64B access and 256B access
- -Tested with 100% reads, 67% reads, 50% reads, 0% reads
- -Tested with both random and sequential access

Flexible IO Tester

- -Tested at 4kB access and 128kB access
- -Tested with 100% reads, 67% reads, 50% reads, 0% reads
- -Tested with **only sequential access**

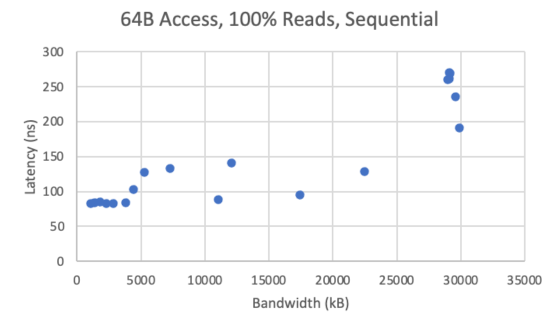
All data is displayed in the form of Latency vs. Bandwidth
(Latency on the y-axis, bandwidth on the x-axis)

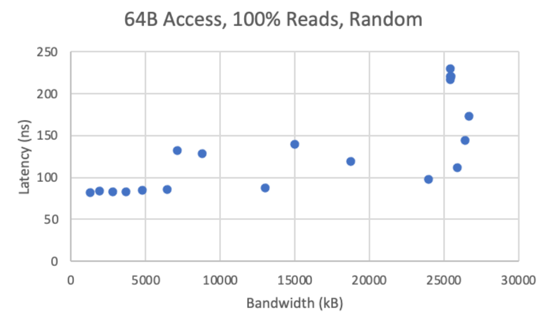
Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

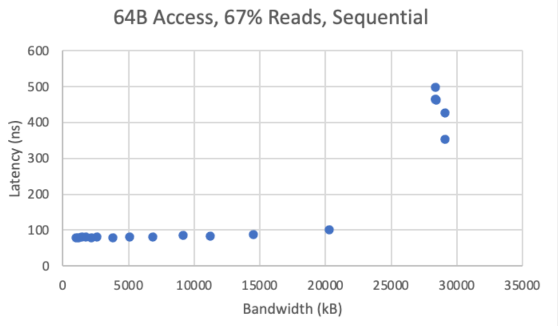
Memory Results:

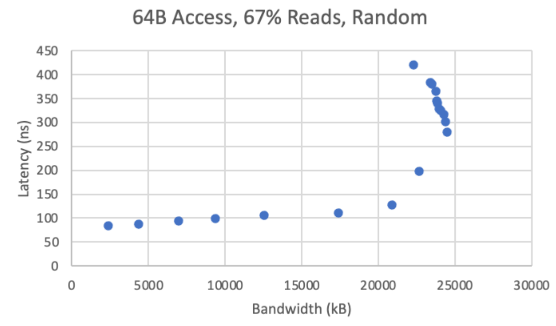
The memory data was obtained using different loaded latencies. To obtain each graph, for various loads, the bandwidth and latency are recorded. Each bandwidth-latency pair is a point on one of the graphs. The results are displayed using bandwidth (in kilobytes) on the x-axis and latency (in nanoseconds) on the y-axis. This process is repeated for every combination measured.

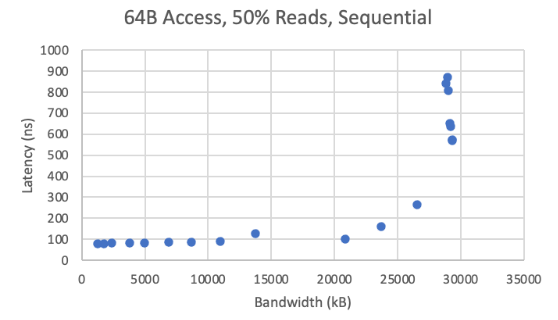
64B Access:

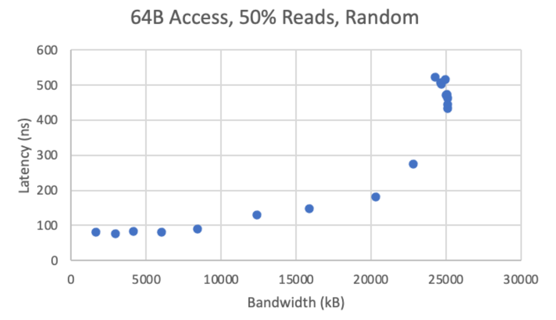

64B Access, 100% Reads, Sequential

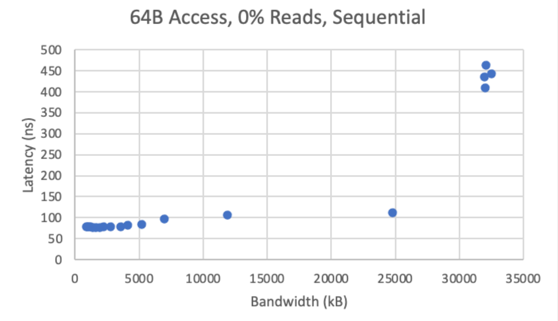

64B Access, 100% Reads, Random

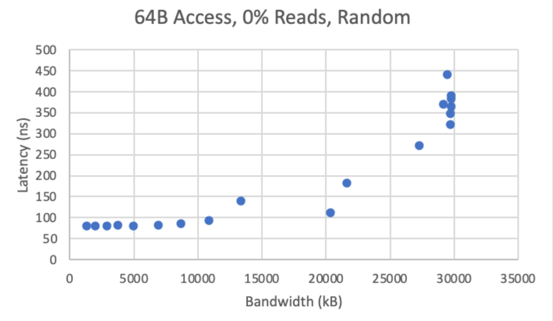

64B Access, 67% Reads, Sequential


64B Access, 67% Reads, Random


64B Access, 50% Reads, Sequential

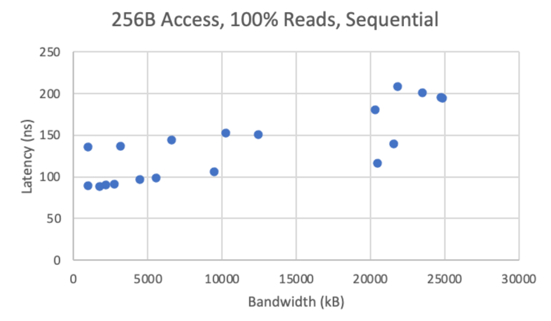

64B Access, 50% Reads, Random

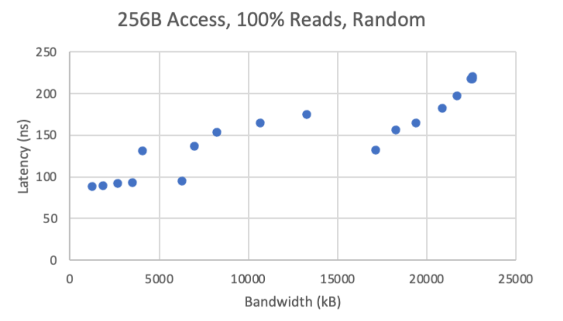

64B Access, 0% Reads, Sequential

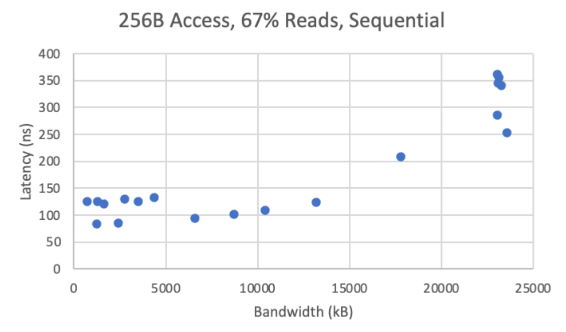

64B Access, 0% Reads, Random

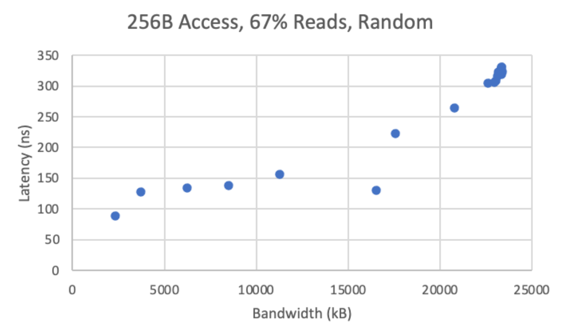
Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

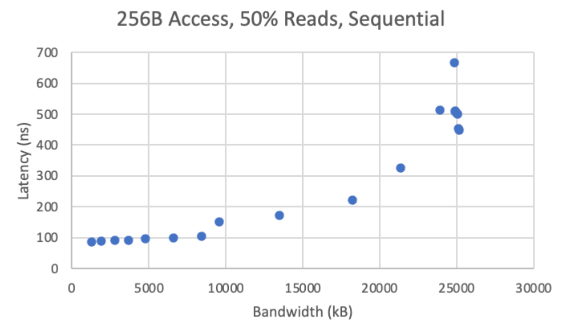
256B Access:

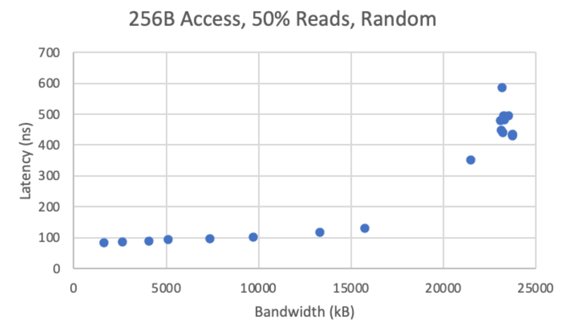

256B Access, 100% Reads, Sequential

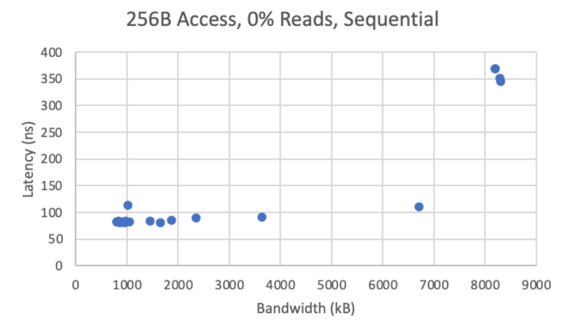

256B Access, 100% Reads, Random

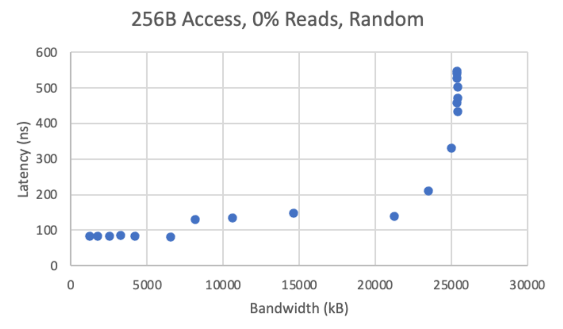

256B Access, 67% Reads, Sequential


256B Access, 67% Reads, Random


256B Access, 50% Reads, Sequential


256B Access, 50% Reads, Random


256B Access, 0% Reads, Sequential


256B Access, 0% Reads, Random

Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

Memory Results Analysis:

There is a clear relationship between the bandwidth and latency in every set of conditions. From all reads to all writes, from 64B access to 256B access, and whether data access is sequential or random, there is always a positive correlation between bandwidth and latency. As bandwidth increases, latency increases exponentially. This relationship matches expectations set by queue theory. To fully utilize the memory's IO capabilities, there must be a large queue of requests. This large queue results in large latencies for each individual request.

In most cases, there is not a significant difference between the sequential access and random access. The sequential access appears to have a slightly higher bandwidth, resulting in a slightly higher latency. The only significant difference appears with 256B access and 0% reads (100% writes), where the bandwidth of the sequential access is far lower. This test was repeated numerous times with the same results, and we are unable to explain the sudden difference.

With respect to read and write ratios, the bandwidth is generally maximized when there are only reads or writes. The latency is generally minimized when there are only reads or writes. When there is a mix of reads and writes, the bandwidth is lower, and the latency is higher. This contrasts with the typical correlation between bandwidth and latency, though it is logical since IO requests will be more efficient when they are all the same type.

There is not a massive difference between the 64B access and the 256B access. The most notable differences are that the 64B access has both higher bandwidth and higher latency. While the 64B access can unexpectedly access data at a higher bandwidth, each individual access with have more latency.

Overall, the typical memory bandwidth is around 25MB/s. The latency for low-bandwidth access appears to be around 100ns, and the latency for high-bandwidth access can be above 800ns.

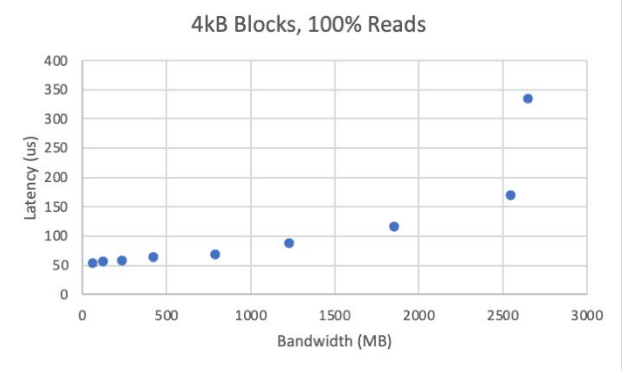
Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

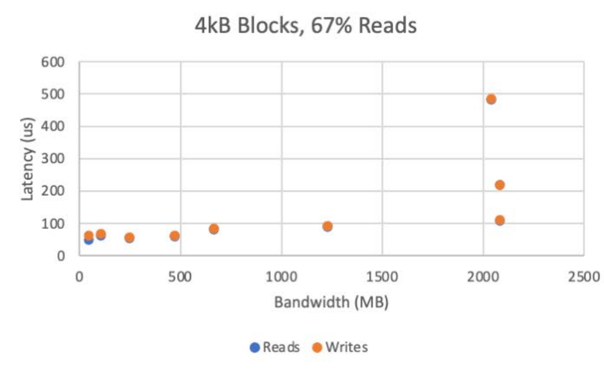
Storage Results:

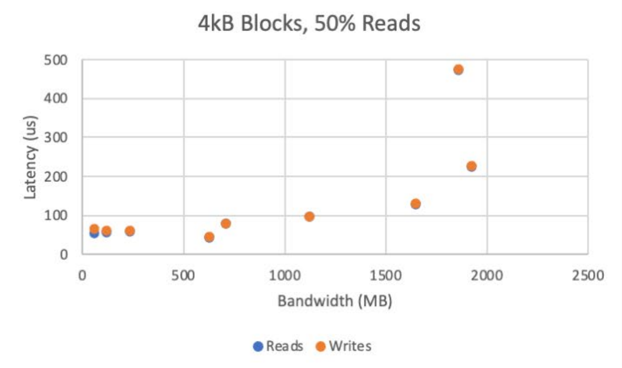
To obtain many bandwidth-latency pairs for the storage, different size bursts of IO requests were made simultaneously. For minimum latency, only one request would be made at a time. For max bandwidth, up to 64 simultaneous requests were made. Each bandwidth-latency pair is a point on one of the graphs. The results are displayed using bandwidth (in megabytes) on the x-axis and latency (in microseconds) on the y-axis. This process is repeated for every combination measured.

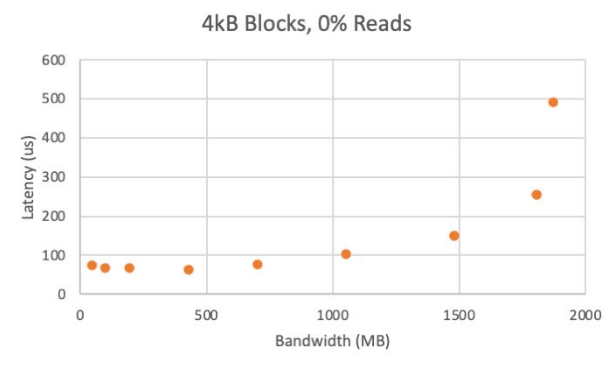
The FIO software displays the read and write latencies separately when there is mixed read/write access. These separate latencies are kept track of in the tables as well, with read latencies denoted in blue and write latencies denoted in orange.

4kB Block Access:



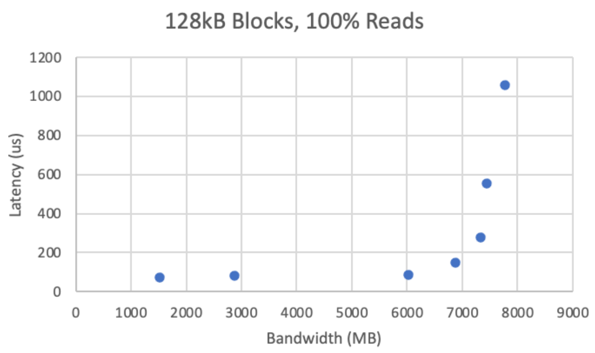


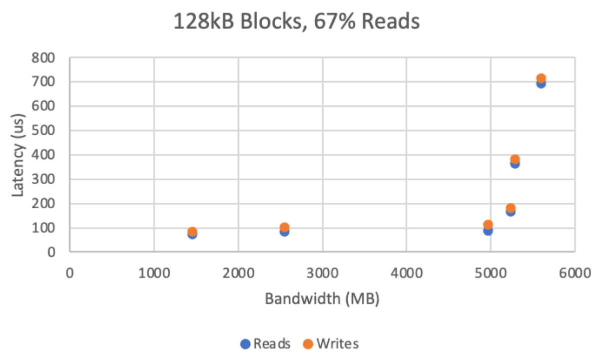


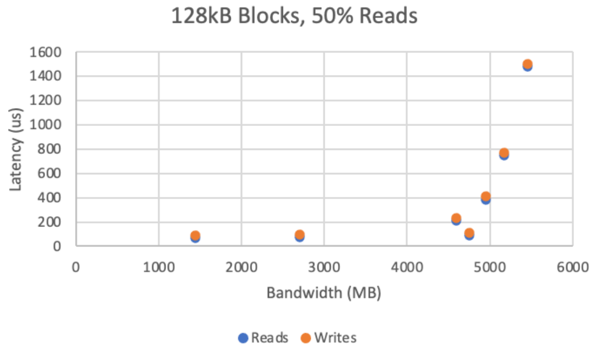


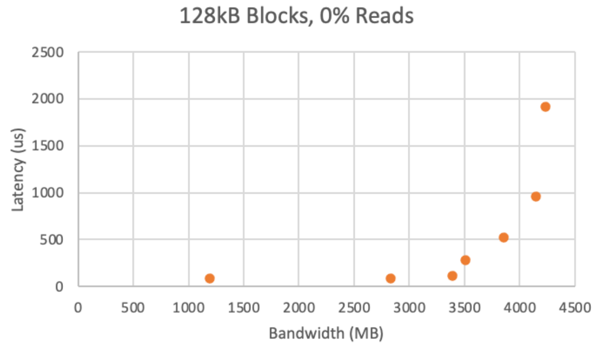
Allan Nesathurai, Bennett Young
Advanced Computer Systems
Project 03

128kB Block Access:









Storage Results Analysis:

Similar to the memory, as the used bandwidth increases, the latency increases exponentially. This holds under all conditions, including both block access size and read/write ratio. The curves look almost the exact same as the ones obtained from the memory data. Latency does not increase too much before a certain bandwidth, but as the bandwidth being used approaches the maximum, the latency shoots up.

Generally, the greater the read/write ratio is, the greater the bandwidth. Also, the greater the read/write ratio, the lower the latency. This is expected because writing to an SSD requires a massive amount of data to be changed. Caching the incoming data helps reduce the penalty but reading large amounts of data is still more time-efficient than writing large amounts of data.

In all cases where there is a mix of reads and writes, the reads have slightly lower latency than the writes. This is not too noticeable, especially as the bandwidth being used increases. The writes take approximately 10us longer than the reads, but this becomes inconsequential as the latency approaches several hundred microseconds.

The storage bandwidth is significantly increased when the data is accessed in 128kB blocks compared to 4kB blocks. This is reasonable because the SSD has quite slow response times for fetching the first piece of data but can fetch many pieces of data at the same time. If more data is requested at once, the bandwidth will be higher. However, accessing the data in larger blocks also increases the latency significantly.

The minimum latency for reading from or writing to the storage is approximately 50us. However, as the bandwidth in use increases, the latency can go above 1ms. The maximum bandwidth using 4kB block access is approximately 2.7GB/s, while the maximum bandwidth using 128kB block access is approximately 8GB/s.

These storage results show a read-only IOPS of approximately 700K and a write-only IOPS of approximately 500K. Intel Data Center NVMe SSD D7-P5600 (1.6TB) has a read-only 4KB IOPS of 400K and write-only 4KB IOPS of 118K. This client-grade SSD shows significantly greater IOPS than the enterprise-grade SSD. One reason for this difference is that enterprise-grade SSDs must be remarkably more reliable. The increased ECC and reliability likely decreases the speed performance.

Conclusion:

There is a tradeoff between bandwidth and latency when utilizing memory or storage. It has been experimentally determined that as the used bandwidth increases, the latency for each request increases significantly. Knowledge of this phenomenon and experience working around it are crucial when designing optimized software.

Intel Memory Latency Checker can be used to determine the specifics of memory bandwidth and latency, and Flexible IO Tester can be used similarly for storage. Programs like these immensely speed up the testing and benchmarking of hardware, as well as give a better idea of what software can be reasonably implemented. The programs can also be used to learn more about memory and storage systems, as well as how to efficiently use them.