nbdX Benchmarking Results Report

HPE NVMe over Fabrics Report

UC Santa Cruz

Software Engineering Senior Design

Authors: John Gemignani, Coy Humphrey,
Eric Litvinsky, Jayden Navarro, Alice Yu

May 5th 2016

Metrics Analysis

We gathered data points for **block sizes of 4K, 32K, 64K and 128K**, and **IO depths of 1, 32, 64, and 128**. We gathered data for both reads and writes, but will focus on the **reads** for our analysis.

Our results were gathered using **fio** and were run by a custom benchmarking framework we developed.

You can find our complete results in CSV format in our GitHub repository under **nvme-hpe -> benchmarking -> results**.

Bandwidth

We found our maximum link speed using the **ib_send_bw** command. The maximum link speed was 37 Gb/s for our QSFP+ interconnect using HPE FDR 40Gb/s network cards.

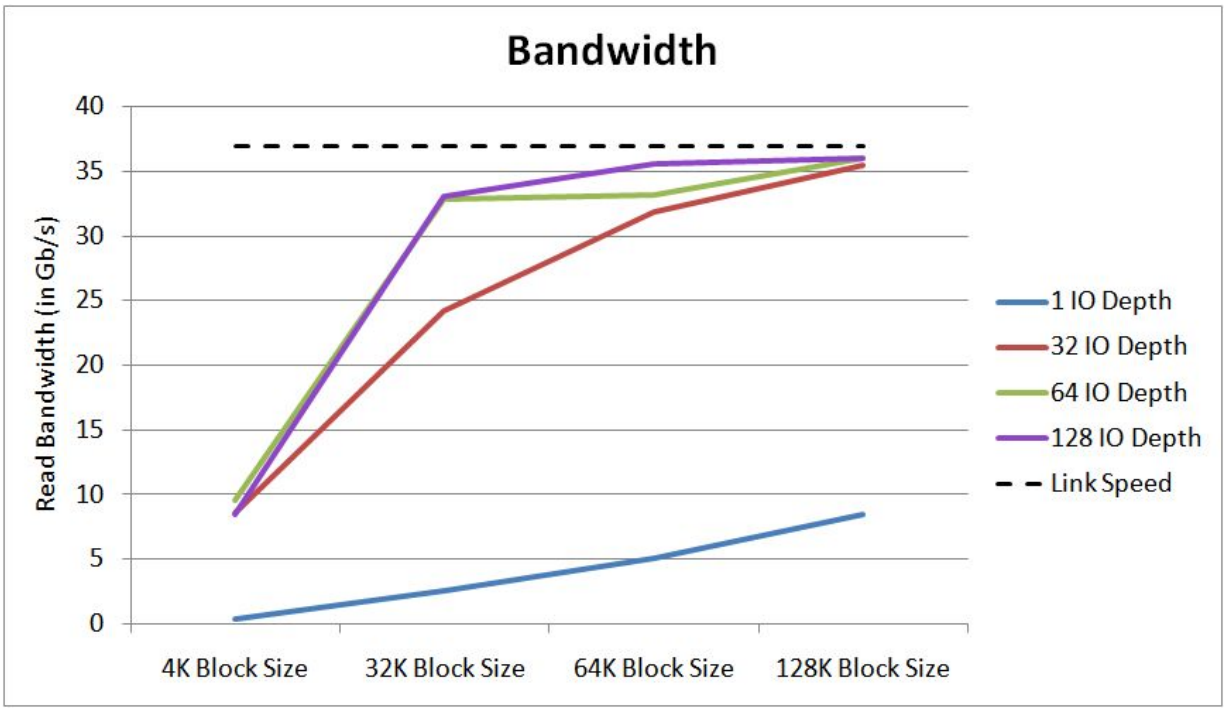


Fig. 1

We found our bandwidth approached our maximum link speed of 37 Gb/s for block sizes 32K, 64K, and 128K, and IO depths 32 and greater, achieving ~36 Gb/s. For an IO depth of 1, which would most likely not be found in a datacenter setting, the bandwidth remained low, but this was to be expected for an under utilized connection. We conclude from these results that the

network link is the bottleneck for transfer throughput, and not the protocol stack. Compared to the 8 Gb/s of current NVMe attached drives [1], our protocol stack should be able to fully support at least four NVMe attached drives with a 40 Gb/s interconnect. Compared to current remote transfer protocols, our latency was respectable, even though it did not achieve the latency goal set out for NVMe over Fabrics.

Latency

We found our card-to-card latency using the **ib_send_lat** command. We found the latency to be 0.77 microseconds for our QSFP+ interconnect using HPE FDR 40Gb/s network cards.

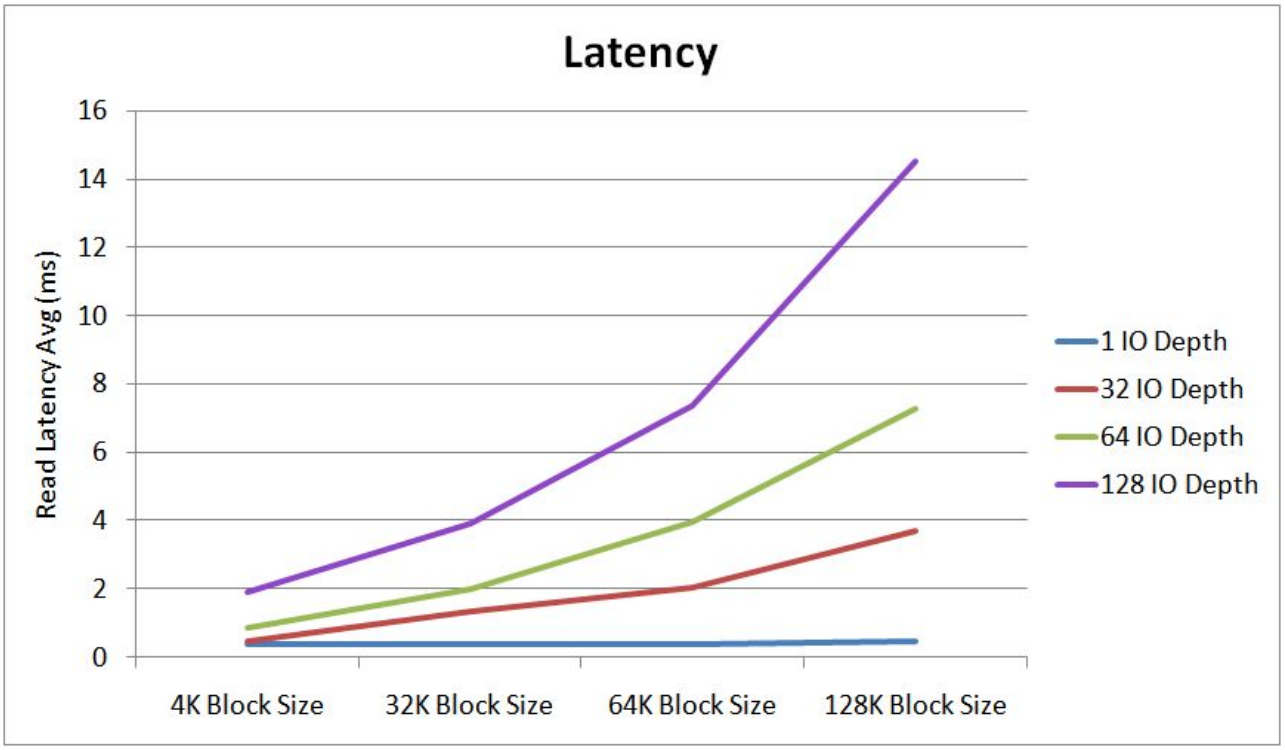


Fig. 2

As the IO depths increased, along with the block sizes, our latency rose into multiple milliseconds, reaching as high as 16 milliseconds. This high latency for large IO depths and block sizes was to be expected as the connection becomes more congested. We found the latency for IO depth of 1 to be ~ 400 microseconds, significantly larger than we were expecting compared to the ~10 microsecond latency specified in the NVMe over Fabrics materials [2].

IOPS

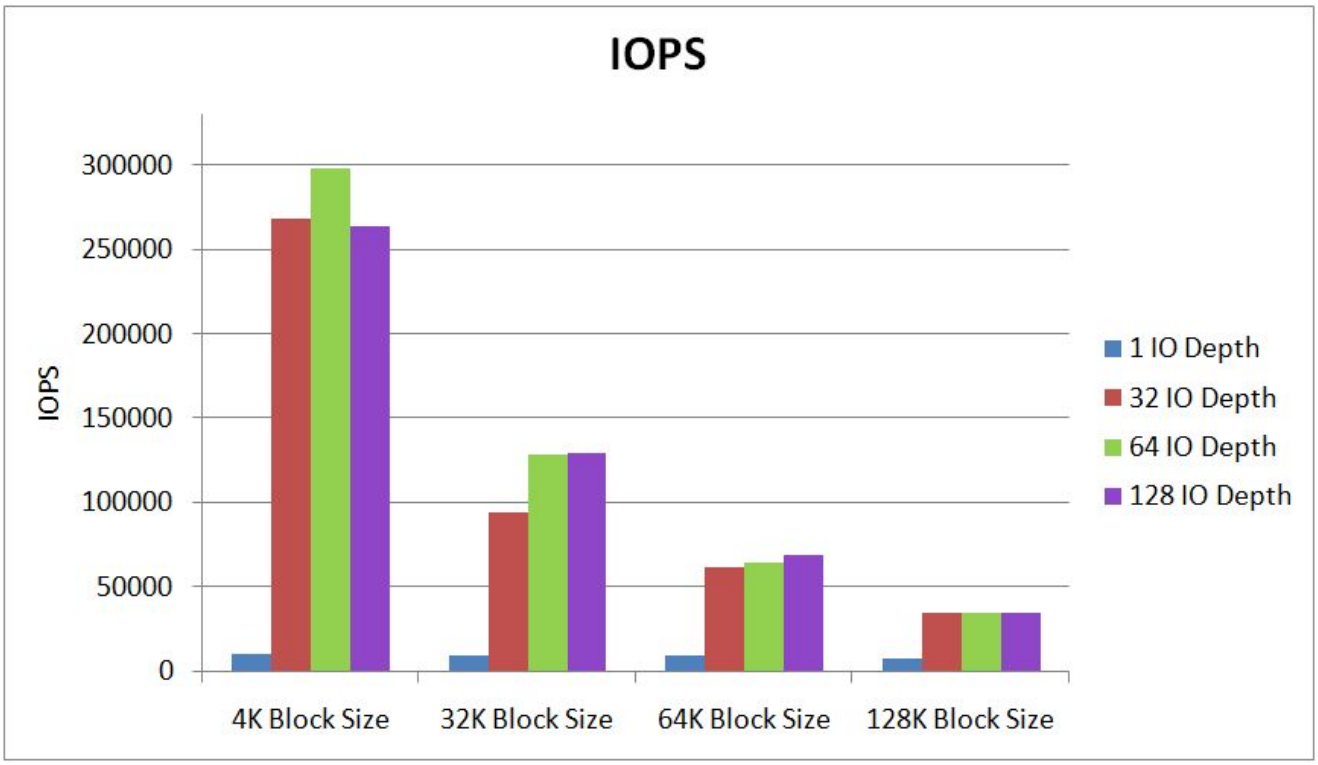
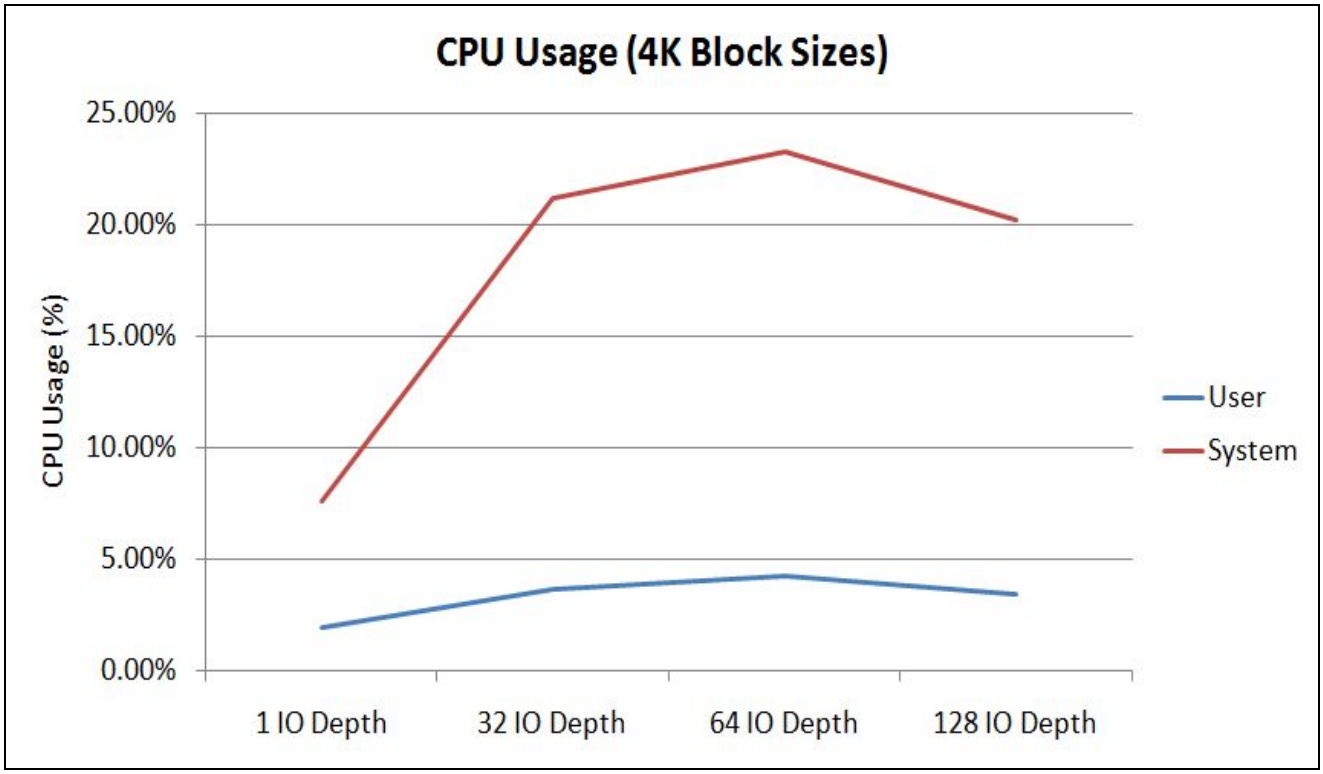


Fig. 3

For IO depths of 32 or greater, as are likely to be found in the datacenter settings where this technology is expected to be used, we found the IOPS for 4k block sizes to fall between 250 thousand and 300 thousand. Compared to the roughly 100 thousand 4k IOPS achieved by current NVMe attached drives [1], our protocol stack can support at least 2 and a half of these drives at full capacity.

CPU Utilization



Here we show our results for 4K block sizes. Larger block sizes used less CPU time.

Assuming that the roughly 5% user CPU usage is from fio generating its workloads, we found that in the worst case nbdx used less than 25% of the system's available CPU time.

We do not expect the CPU utilization of nbdX to be a bottleneck.

Conclusion

In summary, we found the bandwidth approached close to our maximum link speed, our IOPS were more than double that of local NVMe attached drives, and our latency was somewhat larger than what we were expecting, but still within the expected range for a remote transfer protocol.

We conclude that NVMe over Fabrics is a viable solution for network attached storage arrays with NVMe attached drives.

References

[1] http://www8.hp.com/h20195/v2/GetPDF.aspx%2F4AA4-7186ENW.pdf
[2] https://www.openfabrics.org/images/eventpresos/workshops2015/DevWorkshop/Monday/monday_10.pdf