Benchmarking XRootD-HTTPS on 400Gbps Links with Variable Latencies

Aashay Arora[1], Diego Davila[1], Frank Würthwein[1], Justas Balcas[2], Tom Lehman[2], Xi Yang[2] et. al.

1 University of California San Diego
2 Energy Sciences Network (ESnet)

INTRODUCTION

In anticipation of the **High Luminosity-LHC** era, there's a critical need to oversee software readiness for upcoming growth in network traffic for production and user data analysis access.

US-CMS Tier-2 sites need to be able sustain and meet the projected 400Gbps bandwidth demands [1], while tackling the challenge posed by varying latencies between sites.

While we have shown that XRootD can perform at the desired throughput target at low latencies, see figures 1 and 2. In this work we analyze its performance over Long Fat Networks (LFN).

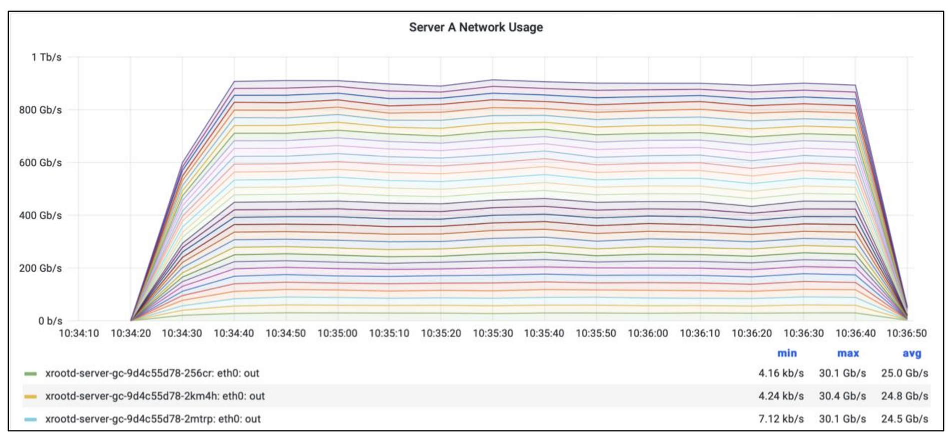


Fig 1. Throughput Graph showing 1 Tbps using XRootD in Microsoft Azure (0.1ms RTT)

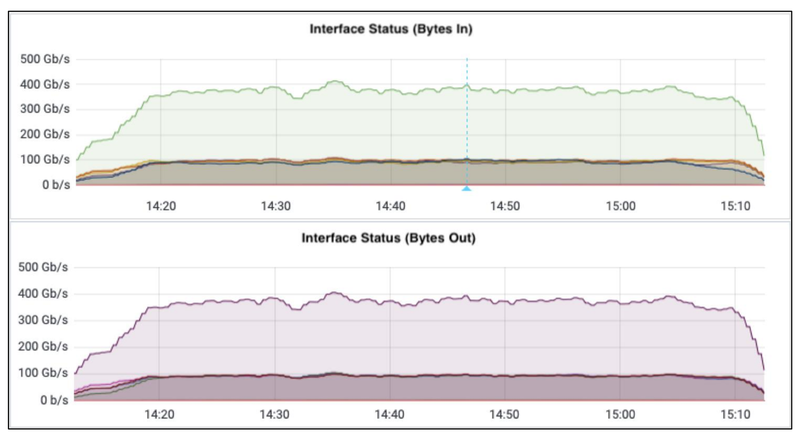


Fig 2. Throughput Graph showing 400 Gbps using XRootD HTTPS between UCSD and Caltech (5ms RTT)

This study focuses on analyzing the **performance of XRootD HTTP TPCs over 400 Gbps links across a variety of latencies using different combinations of cores, XRootD instances and number of active transfers**

Since obtaining geographically distributed hosts connected with high bandwidth links and dedicated for our tests is difficult, we choose to use two identical machines hosted in the San Diego Supercomputer Center and connect them through a variety of real latencies ranging from 5 to 120ms as seen in Figure 3.

TEST SETUP

The hosts used for the tests are identical with the following specifications:

- 2 x 64 Cores @ 3.40 GHz, 1 Ti DDR5 Memory
- Mellanox ConnectX-7 400 Gbps Ethernet

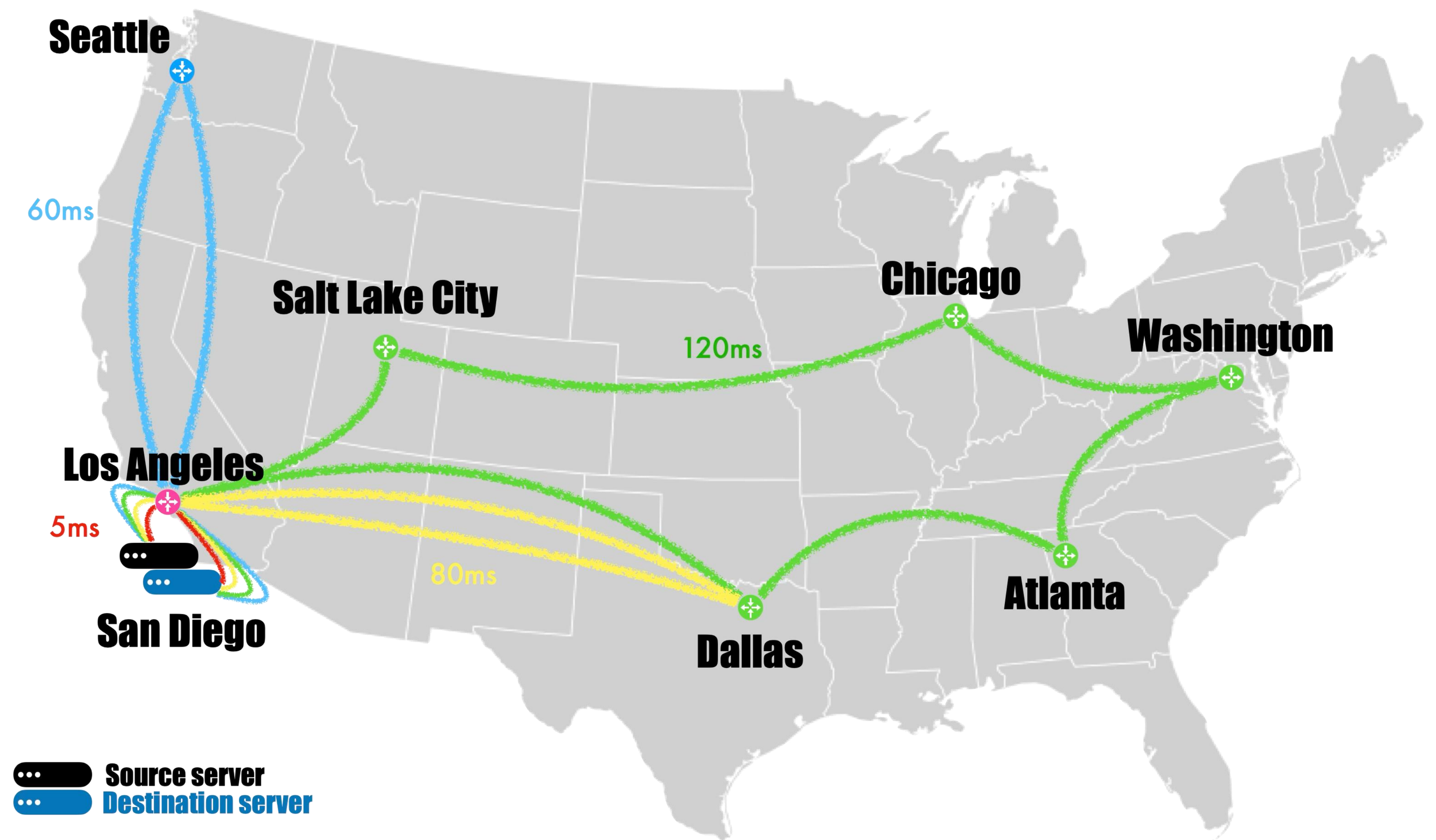


Fig 3. Loops in Fabric with varying Latencies

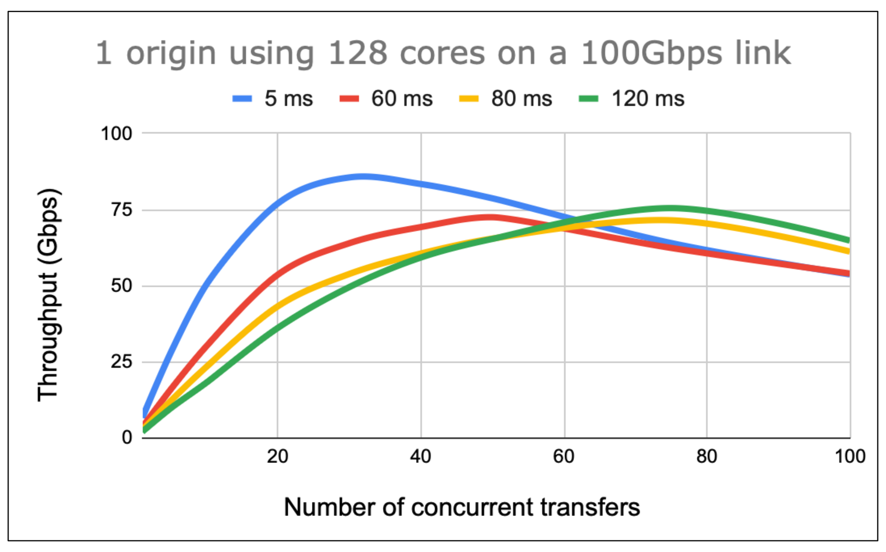
Using SENSE [3] and FABRIC [4] **we routed the traffic between our servers over the WAN and back in a loop-like manner as seen in Figure 3**. This allowed us to perform throughput tests over different latencies using the exact same pair of nodes and doing so over authentic network conditions as opposed to inducing fake latencies with tools like Linux traffic control.

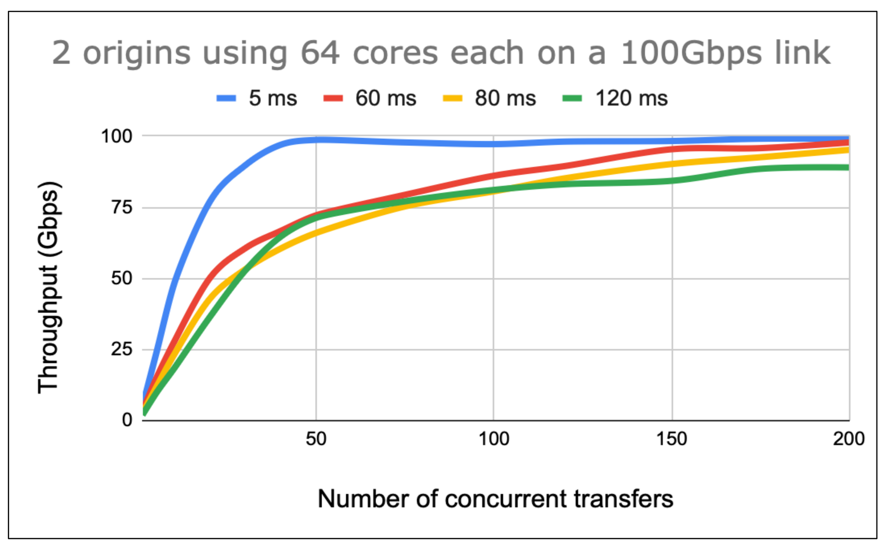
We use Kubernetes to deploy XRootD which eased the management of the number of instances and cores per instance required for our different tests.

RESULTS

By using loops and fixing the bandwidth between the servers we can systematically test different configurations of XRootD deployments to answer questions like

What are the effects of latency on throughput?

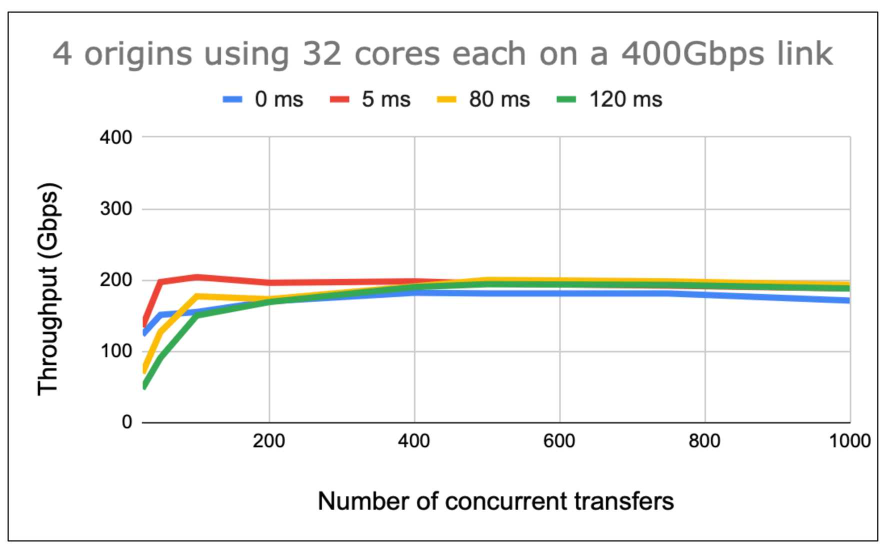


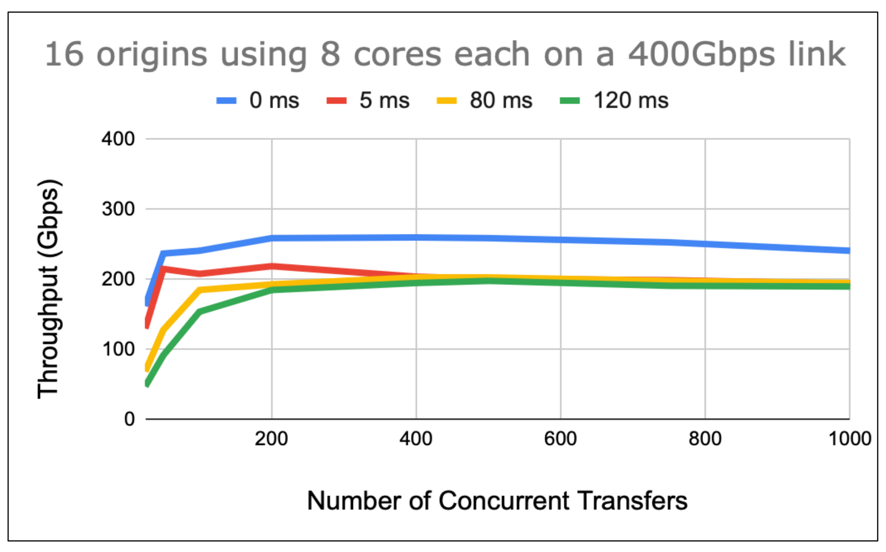


We can see that there is a point in which higher latency give us better throughput. **The higher the latency the slower the drop in performance** as we increase the number of active transfers.

What is the maximum throughput we can get out of a single server?

We have showed in the past [2] that it is possible to reach 400Gbps using many hosts, but what about a single machine?

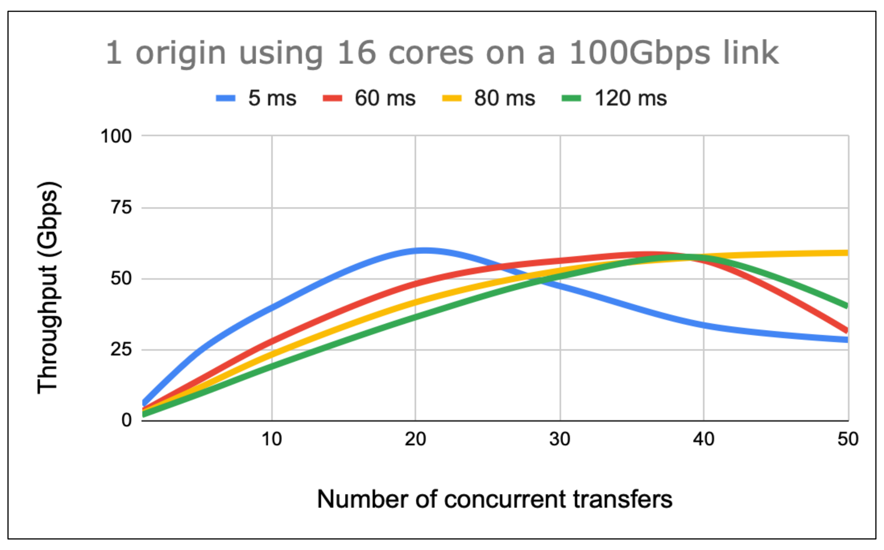


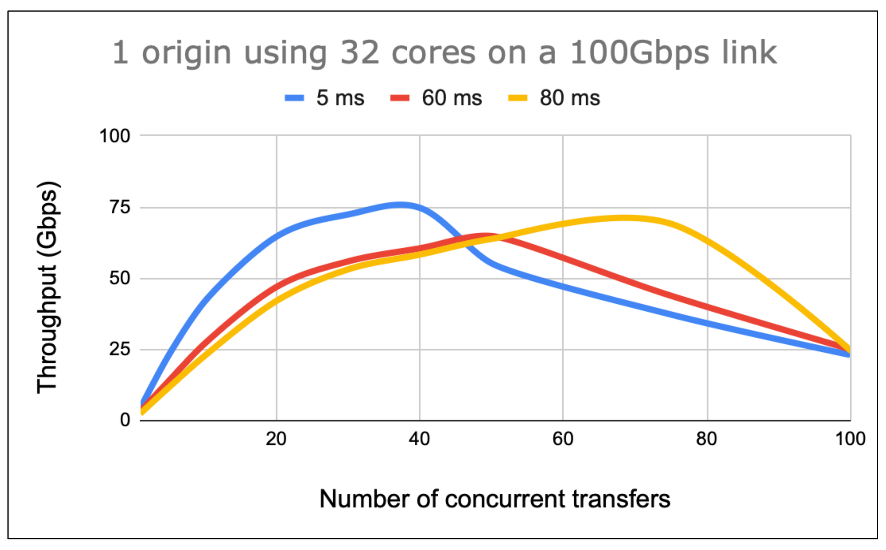


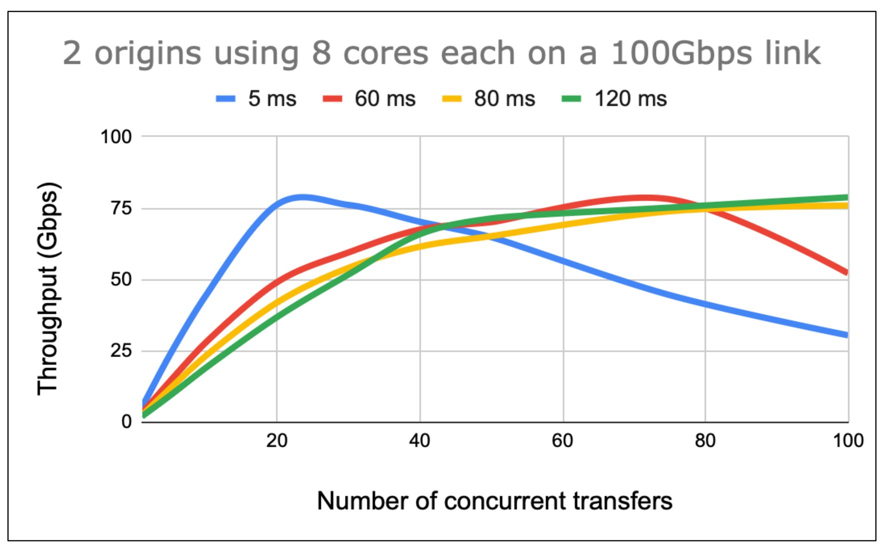
We observe that the maximum we can achieve out of XRootD from a single machine is **260Gbps** which is when the hosts are connected to each other directly (~0ms latency).

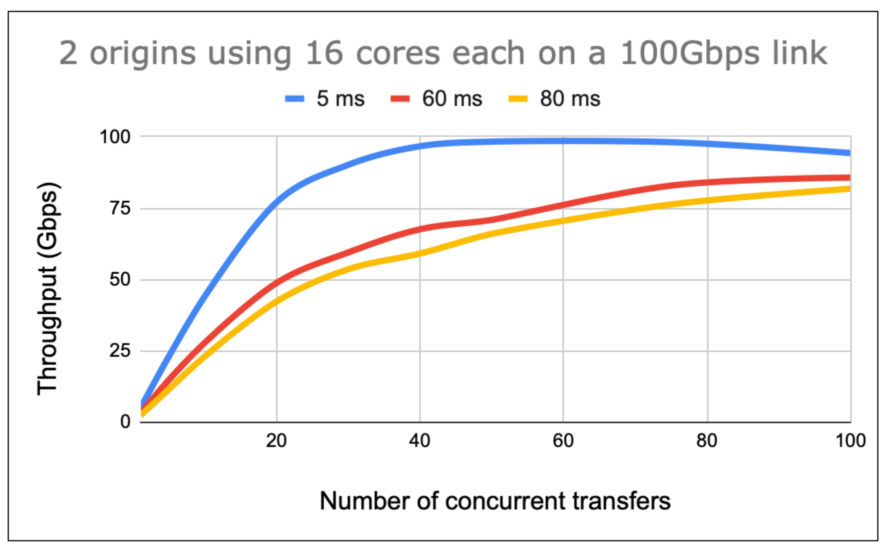
A trend in all the plots is that deploying **more instances of XRootD with fewer cores gives better performance than deploying one instance with many cores**

What is the fewest number of cores and/or instances you need to reach 100Gbps? what about 200Gbps?

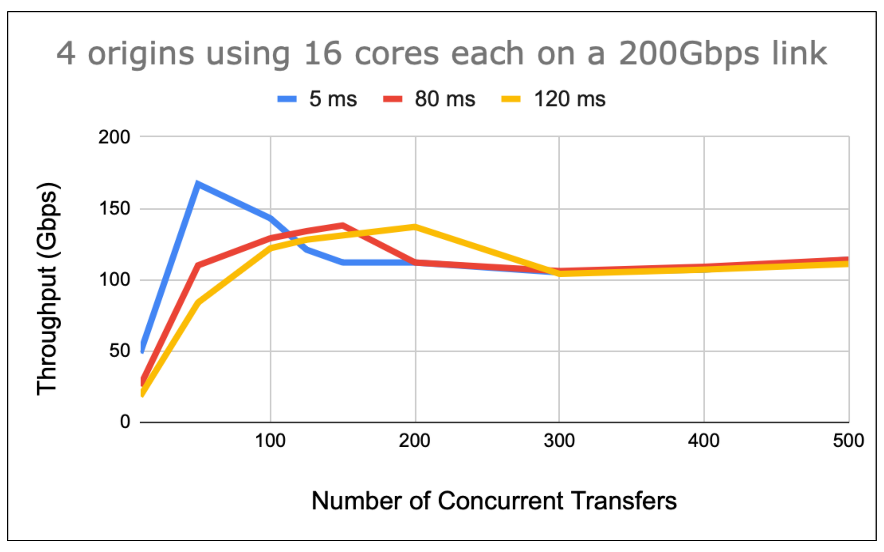


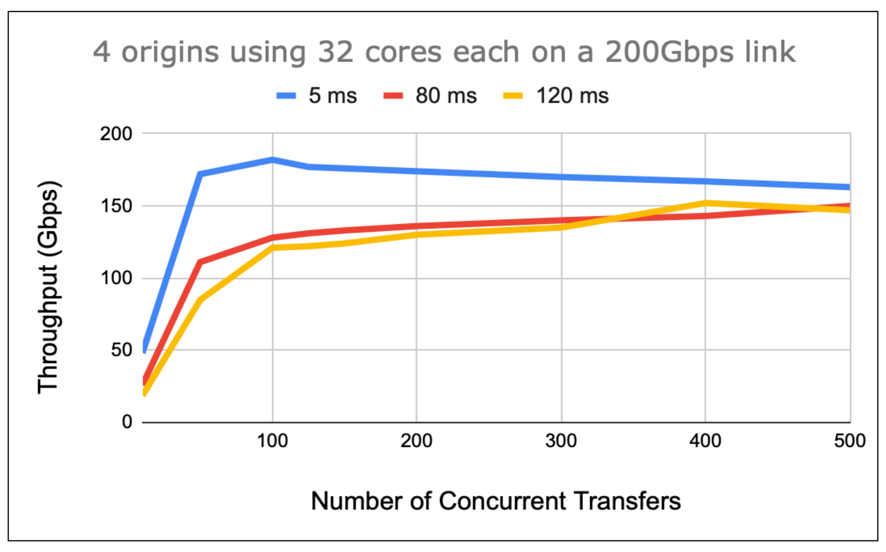






We observe that **a single XRootD instance is not able to saturate a 100Gbps link, 2 origins each using 16 cores can sustain high throughputs even for many concurrent transfers**. For 200Gbps, this is 4 instances using 32 cores





REFERENCES

[1] D. Carder, E. Dart, M. Graf, C. Hawk, A. Holder, D. Jacob, et al., Basic energy sciences network requirements review (final report) (2022) https://escholarship.org/uc/item/3jj0h54n

[2] Arora, Aashay, Guiang, Jonathan, Davila, Diego, Würthwein, Frank, Balcas, Justas, & Newman, Harvey. (2024). 400Gbps benchmark of XRootD HTTP-TPC. EPJ Web of Conf., 295, 01001. doi:10.1051/epjconf/202429501001

[3] Monga, I., Guok, C., MacAuley, J., Sim, A., Newman, H., Balcas, J., … Yang, X. (2020). Software-Defined Network for End-to-end Networked Science at the Exascale. Future Generation Computer Systems, 110, 181–201. doi:10.1016/j.future.2020.04.018

[4] Baldin, Ilya, Anita Nikolich, James Griffioen, Indermohan Inder S. Monga, Kuang-Ching Wang, Tom Lehman, and Paul Ruth. "FABRIC: A national-scale programmable experimental network infrastructure." IEEE Internet Computing 23, no. 6 (2019): 38-47.

ACKNOWLEDGEMENTS

This work is partially supported by the US National Science Foundation (NSF) Grants OAC-1836650, PHY-2323298, PHY-2121686 and OAC-2112167. Finally, this work would not be possible without the significant contributions of collaborators at ESNet, Caltech, and SDSC.