Systematic benchmarking of HTTPS third party copy on 100Gbps links using XRootD

Edgar Fajardo[†], Aashay Arora[†], Diego Davila[†], Richard Gao[†], Frank Würthwein[†], Brian Bockelman[§]

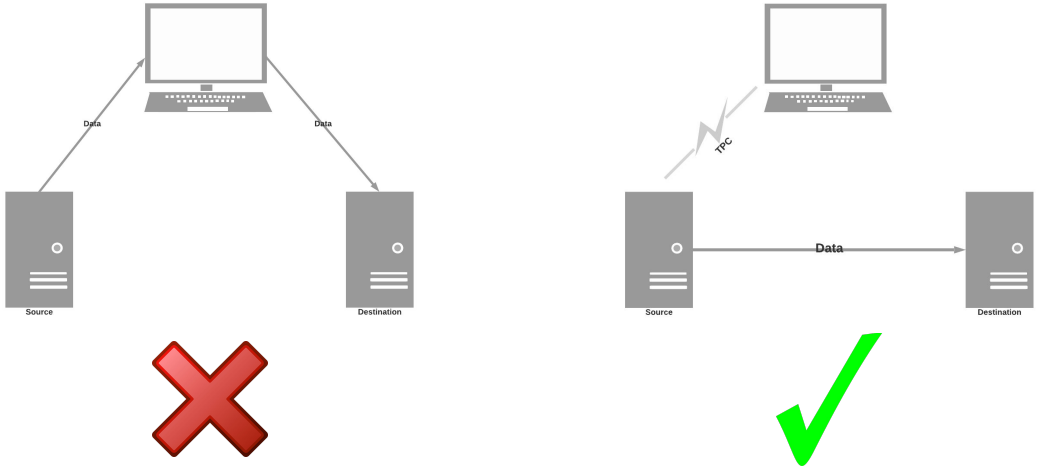
[†]University of California, San Diego; [§]Morgridge Institute for Research

Motivation

- Data produced in LHC is distributed to research sites across the globe using a variety of protocols.

- After the HL-LHC upgrade, we expect to see a thirty fold increase in data storage / transfer needs.
- We explore the scalability of XRootD-HTTPS, a protocol recently adopted within the LHC community as a replacement for the soon-to-be-deprecated GridFTP, which is currently used for the bulk of third party copy (TPC) transfers.

Third Party Copy (TPC)



As opposed to **streaming**, in TPC transfers, a client initiates transfer between two servers. Afterwards only the servers communicate with each other.

Both GridFTP and XRootD-HTTPS support TPC natively.

Objective

- To systematically execute TPC transfers both with GridFTP and XRootD-HTTPS over 100Gbps links and compare their performance.
- Study the sensitivity of these protocols to latency (RTT) between servers.

Methodology Overview

Step 1. Create Docker images for both XRootD and GridFTP using RPMs from OSG[1]

Step 2. Deploy one container on each 100Gbps capable node via Kubernetes, in the Pacific Research Platform (PRP)[2], a worldwide distributed Kubernetes cluster.

Step 3. Run TPC transfers for each pair of nodes using both XRootD-HTTPS, and GridFTP and calculate transfer rates.

Step 4. Create a mesh of measured throughputs using MadDash for both XRootD and GridFTP to monitor performance of these protocols relative to the hardware link (baseline determined using ethr).

Step 1: Container Creation

Starting from OSG Software Base image, we install OSG Standalone RPMs for XRootD and GridFTP to build two Docker images.

```
FROM opensciencegrid/software-base:fresh

RUN yum -y install vim && \
    yum -y install osg-ca-certs && \
    yum -y install osg-xrootd-standalone --enablerepo=osg-upcoming && \
    yum -y install osg-test osg-ca-generator --enablerepo=devops-itb

ADD image-config.d/* /etc/osg/image-config.d/

ADD supervisord.conf /etc/supervisord.d/10-xrootd-https-test.conf
```

XRootD-HTTPS

```
FROM opensciencegrid/software-base:fresh

RUN yum -y install vim && \
    yum -y install osg-ca-certs && \
    yum -y install osg-test osg-ca-generator --enablerepo=devops-itb && \
    yum -y install osg-gridftp && \
    yum -y install gfal2-plugin-file gfal2-plugin-gridftp && \
    yum -y install globus-gass-copy-progs

RUN adduser tpcuser

ADD image-config.d/* /etc/osg/image-config.d/

RUN mkdir -p /var/log/supervisor
ADD supervisord.conf /etc/supervisord.d/10-gridftp.conf

ADD grid-mapfile /etc/grid-security/grid-mapfile
```

GridFTP

Step 2 & 3: Deploying the Pods & Running TPCs

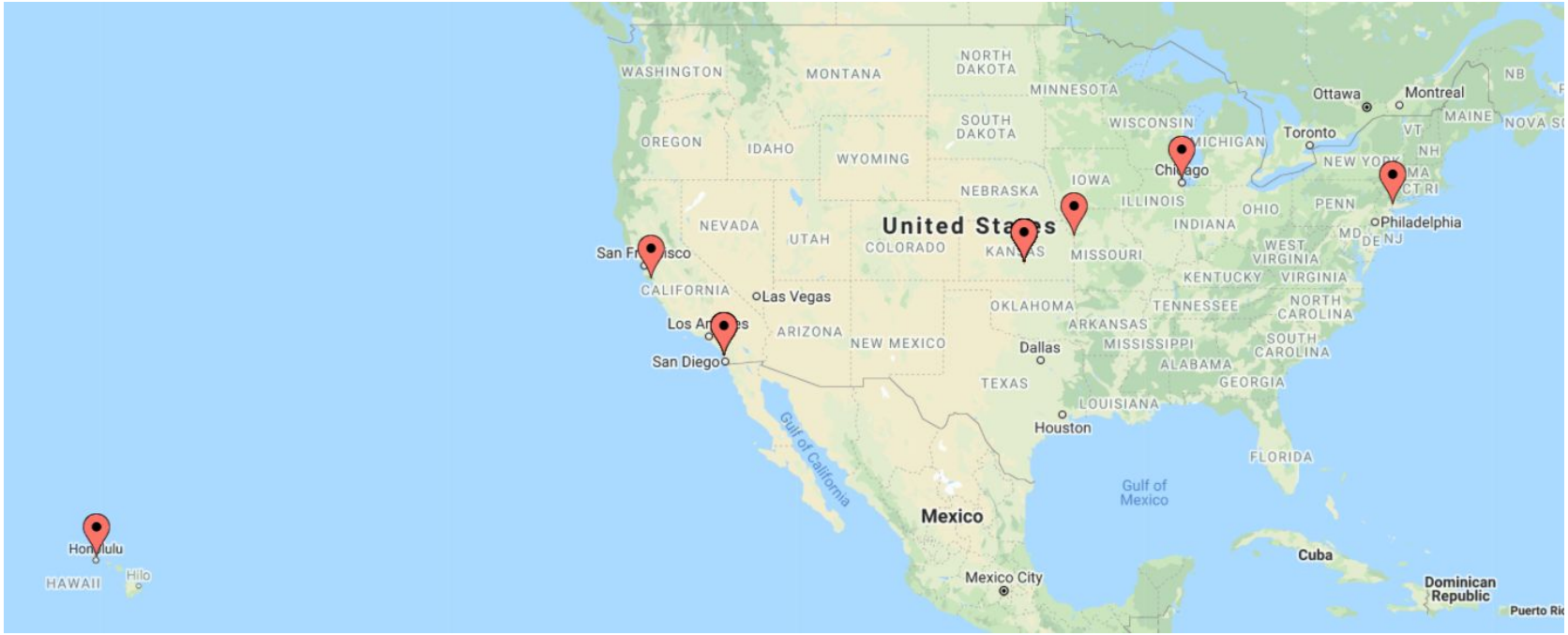
NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
tpc-xrootd-2tscq	1/1	Running	0	73m	10.244.84.47	osg.kans.nrp.internet2.edu
tpc-xrootd-4vwmk	1/1	Running	0	73m	10.244.111.153	k8s-igrok-04.calit2.optiputer.net
tpc-xrootd-548z9	0/1	Pending	0	73m	<none>	<none>
tpc-xrootd-65cb6	0/1	Pending	0	73m	<none>	<none>
tpc-xrootd-9b64g	1/1	Running	0	73m	10.244.208.175	prp01.ifa.hawaii.edu
tpc-xrootd-9d5w8	1/1	Running	0	73m	10.244.1.13	k8s-nvme-01.sdsc.optiputer.net
tpc-xrootd-b2rm2	1/1	Running	0	73m	10.244.130.171	k8s-igrok-07.calit2.optiputer.net
tpc-xrootd-bxbkp	0/1	Pending	0	73m	<none>	<none>
tpc-xrootd-clgrc	1/1	Running	0	73m	10.244.115.81	k8s-igrok-05.calit2.optiputer.net
tpc-xrootd-gcm95	1/1	Running	0	73m	10.244.90.99	ps-100g-scidmz-0.tools.ucla.net
tpc-xrootd-gg4bd	1/1	Running	0	73m	10.244.12.11	k8s-epyc-01.sdsc.optiputer.net
tpc-xrootd-hdjkx	1/1	Running	0	73m	10.244.113.232	k8s-igrok-06.calit2.optiputer.net
tpc-xrootd-hk2nh	0/1	Pending	0	73m	<none>	<none>
tpc-xrootd-kflk1	1/1	Running	0	73m	10.244.14.52	ps-100g.sdsu.edu
tpc-xrootd-l29hd	1/1	Running	0	73m	10.244.216.108	k8s-nvme-01.ultralight.org
tpc-xrootd-lcmj2	0/1	Terminating	0	5d8h	<none>	k8s-gen4-01.calit2.optiputer.net
tpc-xrootd-llvfz	1/1	Running	0	73m	10.244.71.186	fiona-100g.ucsc.edu
tpc-xrootd-ptlfv	1/1	Running	0	73m	10.244.154.31	k8s-igrok-01.calit2.optiputer.net
tpc-xrootd-rbtj7	1/1	Running	0	73m	10.244.11.36	siderea.ucsc.edu
tpc-xrootd-rw69d	1/1	Running	0	73m	10.244.89.60	osg.newy32aoa.nrp.internet2.edu
tpc-xrootd-skff6	1/1	Running	0	73m	10.244.201.197	k8s-igrok-03.calit2.optiputer.net
tpc-xrootd-t5txm	1/1	Running	0	73m	10.244.74.29	osg.chic.nrp.internet2.edu
tpc-xrootd-tlkch	1/1	Running	0	73m	10.244.116.242	k8s-u280-01.calit2.optiputer.net
tpc-xrootd-ww5vf	1/1	Running	0	73m	10.244.90.48	k8s-igrok-02.calit2.optiputer.net
tpc-xrootd-x8hvp	1/1	Running	0	73m	10.244.80.187	maserati.sciencedmz.nps.edu
tpc-xrootd-zndmm	1/1	Running	0	73m	10.244.135.87	stashcache.t2.ucsd.edu
tpc-xrootd-zpxt7	1/1	Running	0	73m	10.244.39.174	nrp-s1.nysernet.org
xrootd-tpc-master-7bf9489988-k4zmp	1/1	Running	0	9h	10.244.254.156	osg-houston-stashcache.nrp.internet2.edu

Transfer Pods

Master Pod

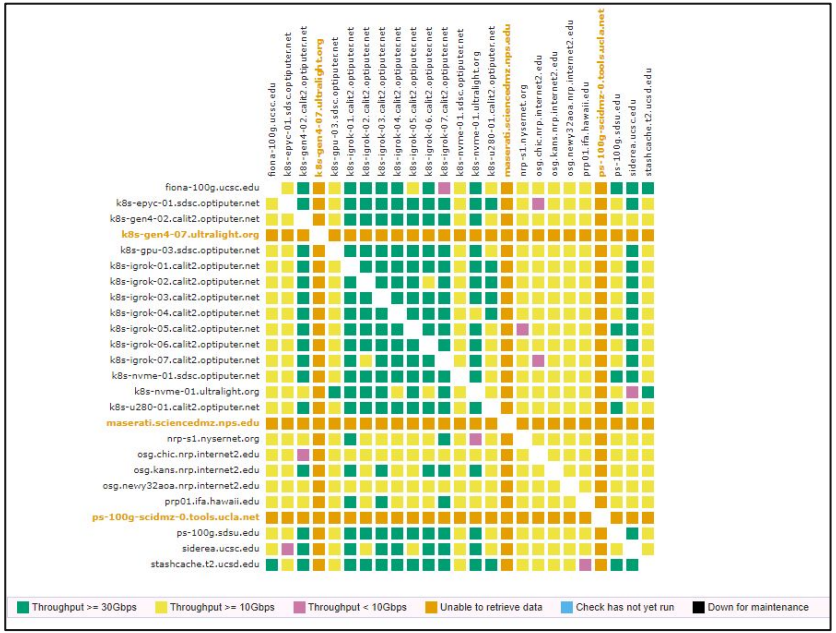
- One pod is deployed on each 100Gbps capable node using a DaemonSet [3]
- TPCs are initiated using the master pod and rates are calculated using a Python script.

Geographical Locations of Nodes Used



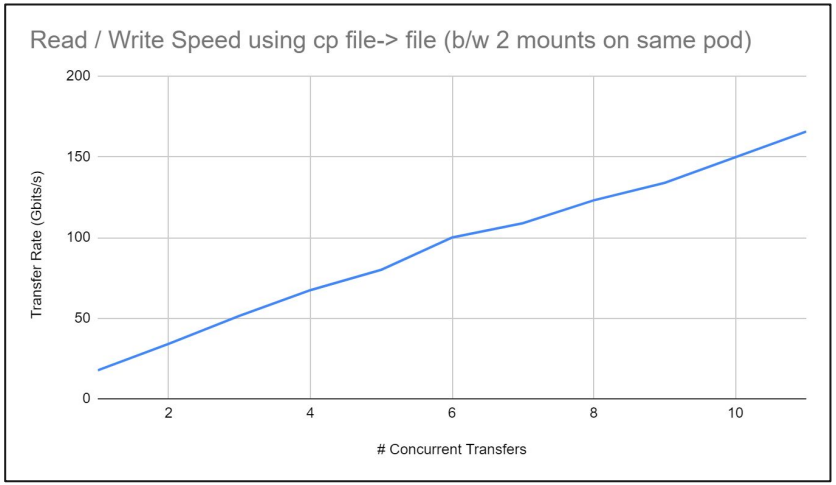
Step 4: Monitoring the Transfers

- TPCs between each pair of nodes are run (sequentially to ensure no node was being overused) and throughput measurements are sent to a Maddash mesh



Fine Tuning each Transfer

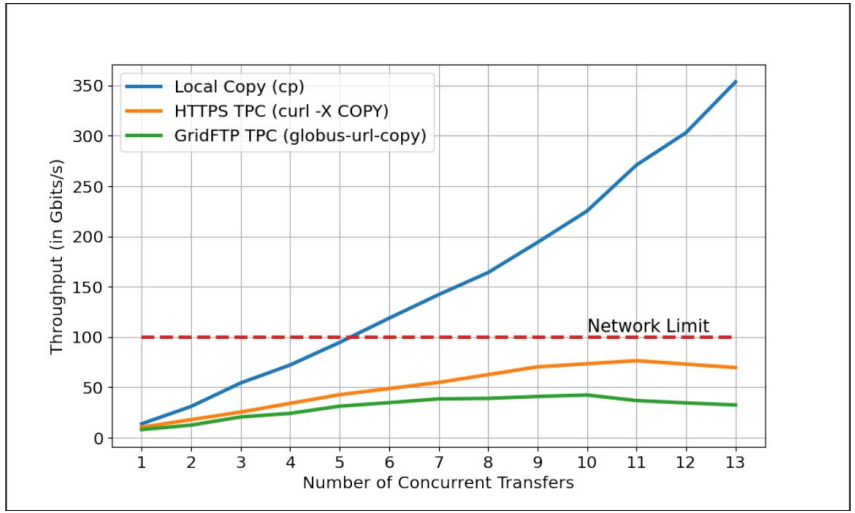
- **Estimating limits of our Infrastructure**
  - Test transfers between two volume mounts in the same pod to understand hardware limitations.
  - Check local read/write speed by making a pod with two memory mounts and copying (cp) files (5GB) between the mounts concurrently.



Six concurrent transfers are enough to reach 100Gbps.

Throughput for more than 6 concurrent transfers should only depend on the XRootD + Network.

- Repeat the test with similar arrangement for XRootD-HTTPS and GridFTP.
  - Objective : To understand the penalty of using the XRootD.



Conclusion: Software introduces an overhead however, we are suspicious that it may be due to the transfers being routed through the NIC and not having enough CPU available (under investigation).

Regardless of the cause, 6 isn't the magic number for XRootD, **we achieve a max. transfer rate of ~ 82 Gbps at 11 concurrent transfers**

Results

- Comparing HTTPS with GridFTP and ethr [4] :

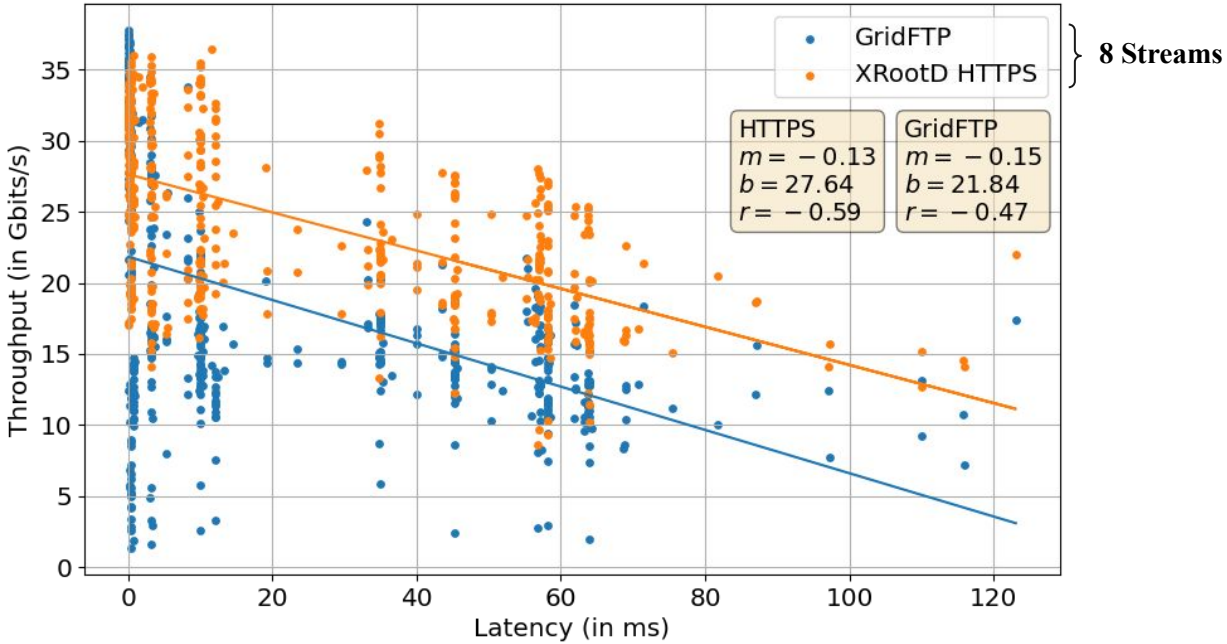
(All tests showcased are using 8 Streams to maintain uniformity)

TPC transfer throughput between two nodes in the same compute room (latency ~0.3 ms) measured over a period of two weeks.

XRootD-HTTPS TPCs slightly outperforms GridFTP, and both perform below fifty percent of the total link capacity measured by ethr (used to determine the baseline)

Throughput as a Function of Latency

After taking measurements over three weeks, we can see the sensitivity of throughput as a function of latency



XRootD-HTTPS consistently outperforms GridFTP on average by ~ 6Gbps or 30%

Both protocols are equally sensitive to latency.

Conclusions

- HTTPS consistently overperforms GridFTP (about 30% better transfer rates on average) given same setup i.e.
  - Network capability
  - Final storage (NVMe or memory)
- We were able to achieve upto 45 Gbps using XRootD-HTTPS but we are still significantly below the throughput of the same hardware using ethr.
  - ethr achieves roughly 80% of the theoretically achievable bandwidth of the hardware link.
- Throughput decreases significantly as we increase the latency.

Acknowledgements

- National Science Foundation through the following grants:
  - PRP: OAC-1541349
  - TNRP: OAC-1826967
  - IRIS-HEP: OAC-1836650
  - OSG: MPS-1148698
- The UCSD Physical Science department and donors for the Undergraduate Summer Research Award ‘20

Questions

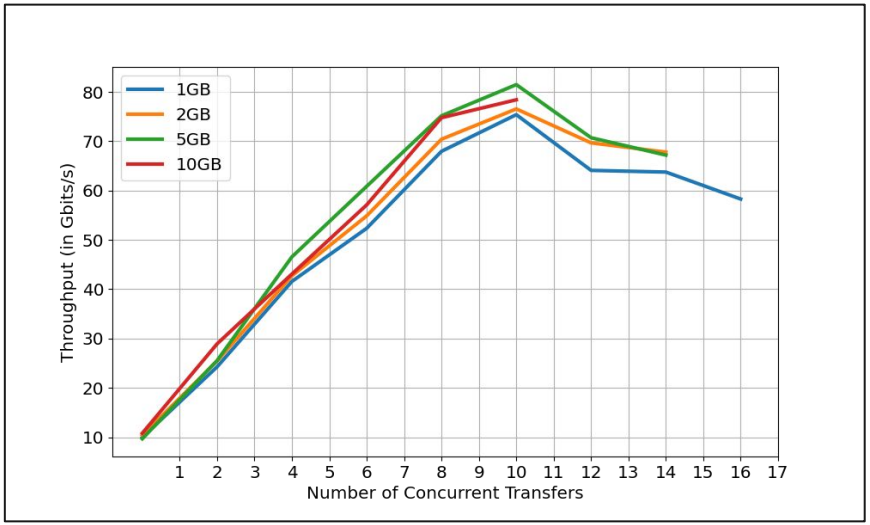
Backup

Why Use This Method?

1. The Pacific Research Platform (PRP) runs a worldwide distributed Kubernetes cluster with several 100 Gbps capable links.
2. Kubernetes allows for very high scalability, pods can be deployed on any number of nodes
3. Transfer Rate can be calculated easily by doing transfers between pods using the node's host network.
4. Tests can be replicated by building from my image on any cluster.

Tuning the TPC Transfers

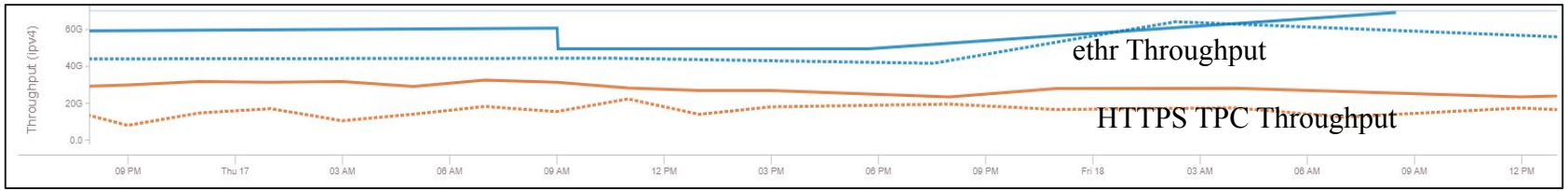
TPCs are tested using various file sizes to estimate the optimal size for maximizing throughput and at the same time, requiring minimum resources.

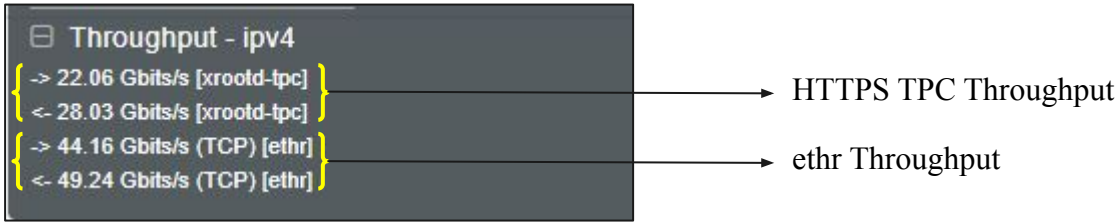


Transfer Rate does **not** change (significantly) with file size, therefore 1 GB files are chosen.

Nodes far Apart

For stashcache.t2.ucsd.edu → osg.kans.nrp.internet2.edu,





These nodes happen to be more than 1k miles apart (45 ms)