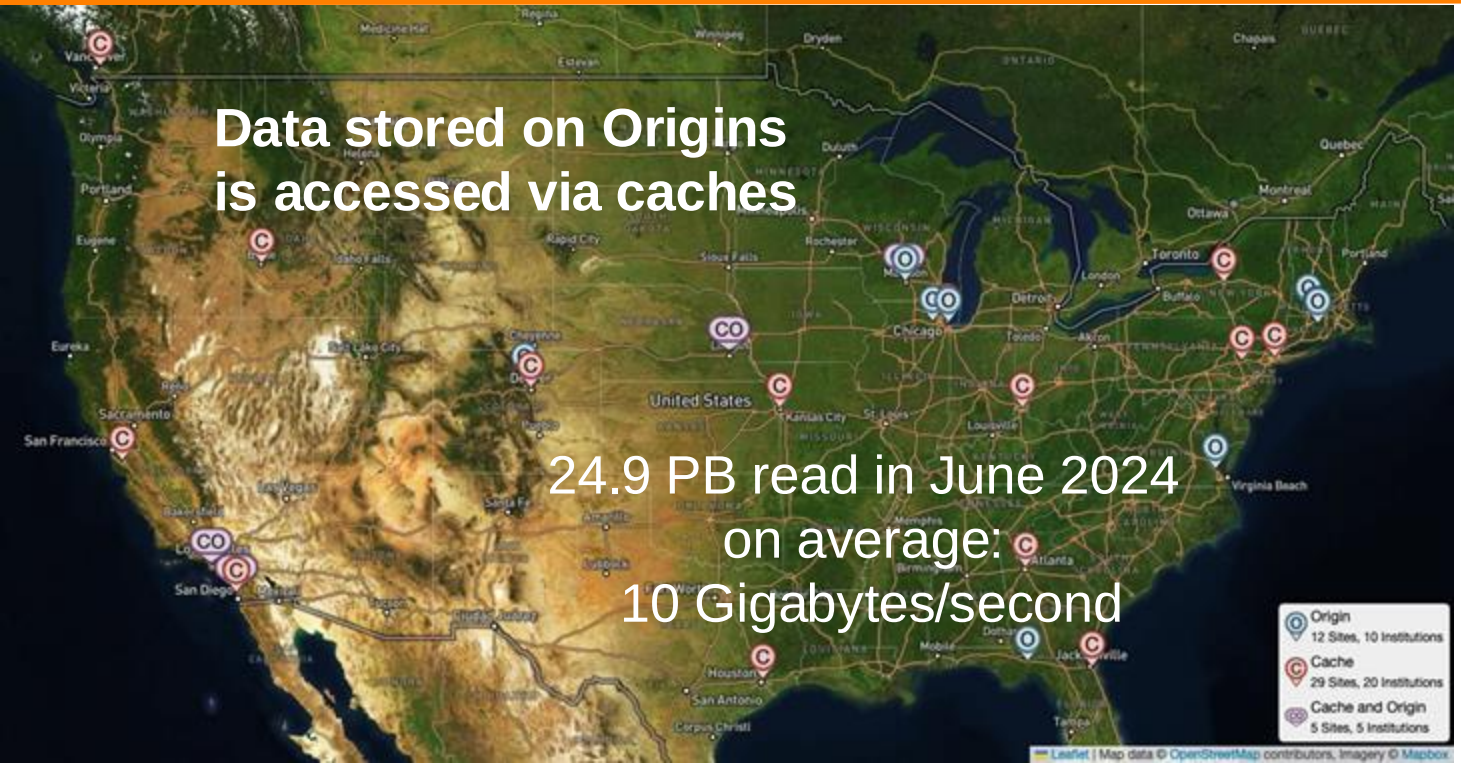
Benchmarking OSDF services to develop XrootD best practices

Fabio Andrijauskas - fandrijauskas@ucsd.edu
Igor Sfiligoi
Frank Wurthwein

University of California San Diego

Zoomed in on Continental USA



80 Gigabit per second … that's 80% of a 100G pipe

Observe <3% cache misses => OSDF caches save >75Gbps in network traffic

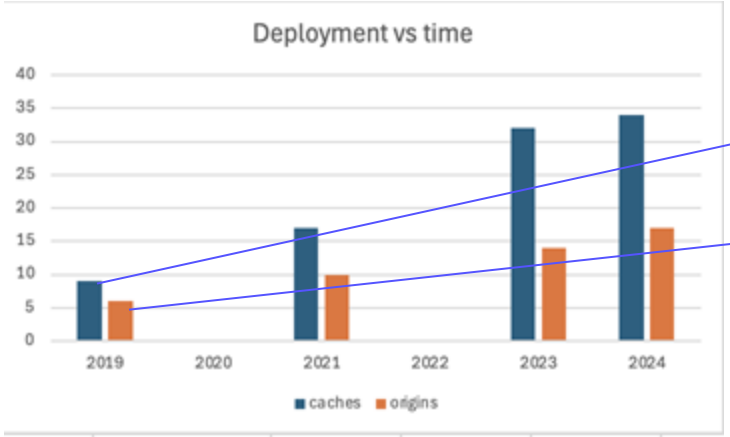
OSG

UCSD

OSDF by Numbers

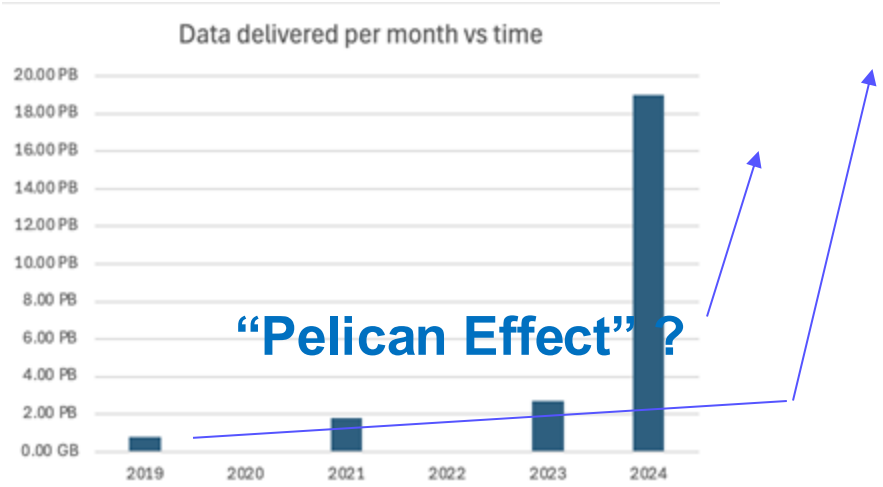
Realtime visualization at:
https://osdf.osg-htc.org

Historic Perspective



~5 caches added per year
~2 origins added per year

Data volume delivered per month
went from ~40% growth per year
between 2019 – 2023 to
7x growth in the last year



Fun Facts for the Month of June

24.9 PB read total

10% of this is accounted for by the OSPool

June 2024 OSPool Numbers

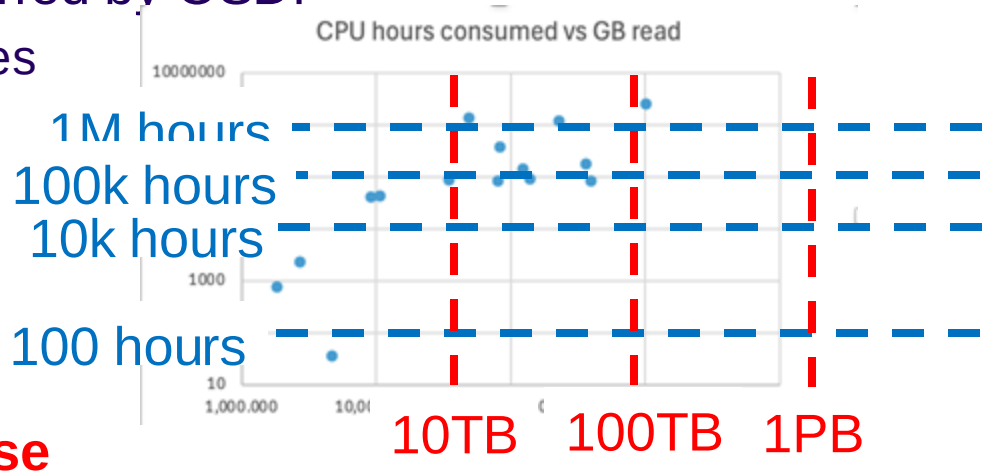
- 61 out of 172 users used OSDF
- 31 out of 98 projects used OSDF

~ 1/3 of OSPool uses OSDF !!!

- OSPool users transfer small files with HTCondor and large files with OSDF:
  - 43% of all bytes transferred by OSDF
    - But only 2.3% of all files

About a dozen projects read 10TB to 1PB consuming 10k to 1M CPU-h during the month of June

Data use is only very loosely correlated with CPU use



OSDF DevOps

- Close to 40 hosts across more than 35 institutions.
- Close to 50 pods.
- 3 deployments models:
  - K8S
  - Docker
  - RPM


FEAR

Why

- What are the XrootD limits?
- How many streams are necessary to use an xGbs network fully?
- What is the best configuration for XrootD using the x HW?
- How much “cost” to use K8S?

Each item has an impact (1 to 5) and complexity (1 to 5) to be completed.

The impact is related to improving the service availability, security, new features, user experience, etc.

The complexity is related to the required time to create the test case and the time to run the time.

File transfer from an origin to a client.

Impact: 5

Complexity: 3

Description. Test the transfer rate using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 8, 32, 64, 128 to N streams (where N is the number of parallel transfers) in the LAN and the WAN, using at least three significantly different RTT values.
Check throughput for various RTTs and some async settings. This will inform us if we should make this more configurable, either through opaque parameters or automatically, based on detected RTT also, check the transfer rate using different clients (wget, curl, pelican) and HTCondor jobs. This set of tests should be able to create a transfer rate base.

How XrootD behaves with a High load on CPU and IO for 24 hours.

Impact: 3

Complexity: 3

Description Check the number of errors or problems on the logs and on the requests with file requests using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB), document how the main storage is mounted, check IO load, and other software configurations.

Check the best resource configuration between K8S Pod, host resources, and XrootD parameters.

Impact: 2

Complexity: 3

Description: Checking the balance between the host resources and the POD resources using different kinds of tests.

Check the transfer rate difference between an origin access and the closest cache

Complexity: 3

Impact: 3

Description. Check the transfer rate difference between origin access and the closest cache, using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) and test the evict function on the cache.

Check the transfer rate related to the storage type.

Complexity: 3

Impact: 5

Description Test the transfer rate using SSD and HDD (RAID and accessing the driver directly) using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 2, 4, and 8 streams.

Check the transfer rate between an authenticated and unauthenticated

Complexity: 3

Impact: 5

Description. Check the transfer rate between an authenticated and unauthenticated, using tokens, CVMFS (auth or not), or certificate and using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 2, 4, and 8 streams.

Check the overhead of tokens

Complexity: 3

Impact: 5

Description. Check the overhead of tokens, generate X unique tokens to avoid cache and see how quickly XRootD can authorize them.

Check the transfer rate using HTTP Third party copy.

Complexity: 3

Impact: 5

Check the transfer rate using HTTP Third party copy using tokens or certificates and using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 2, 4, and 8 streams.

Check the performance of EL7 vs EL9 as K8S OS POD.

Complexity: 3

Impact: 4

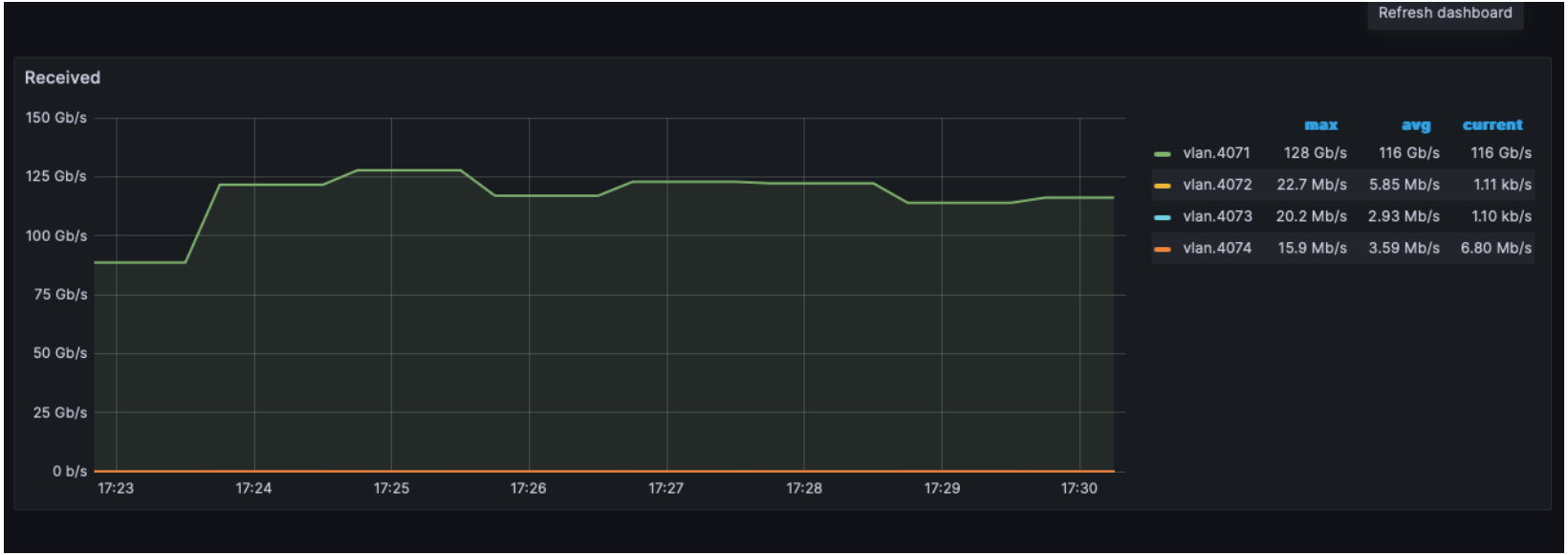
Description. Test the transfer rate using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 2, 4, and 8 streams and EL7 vs EL9 as K8S OS POD.

Check how the redirector performs, as well as the architecture, and improve the monitoring.

Complexity: 3

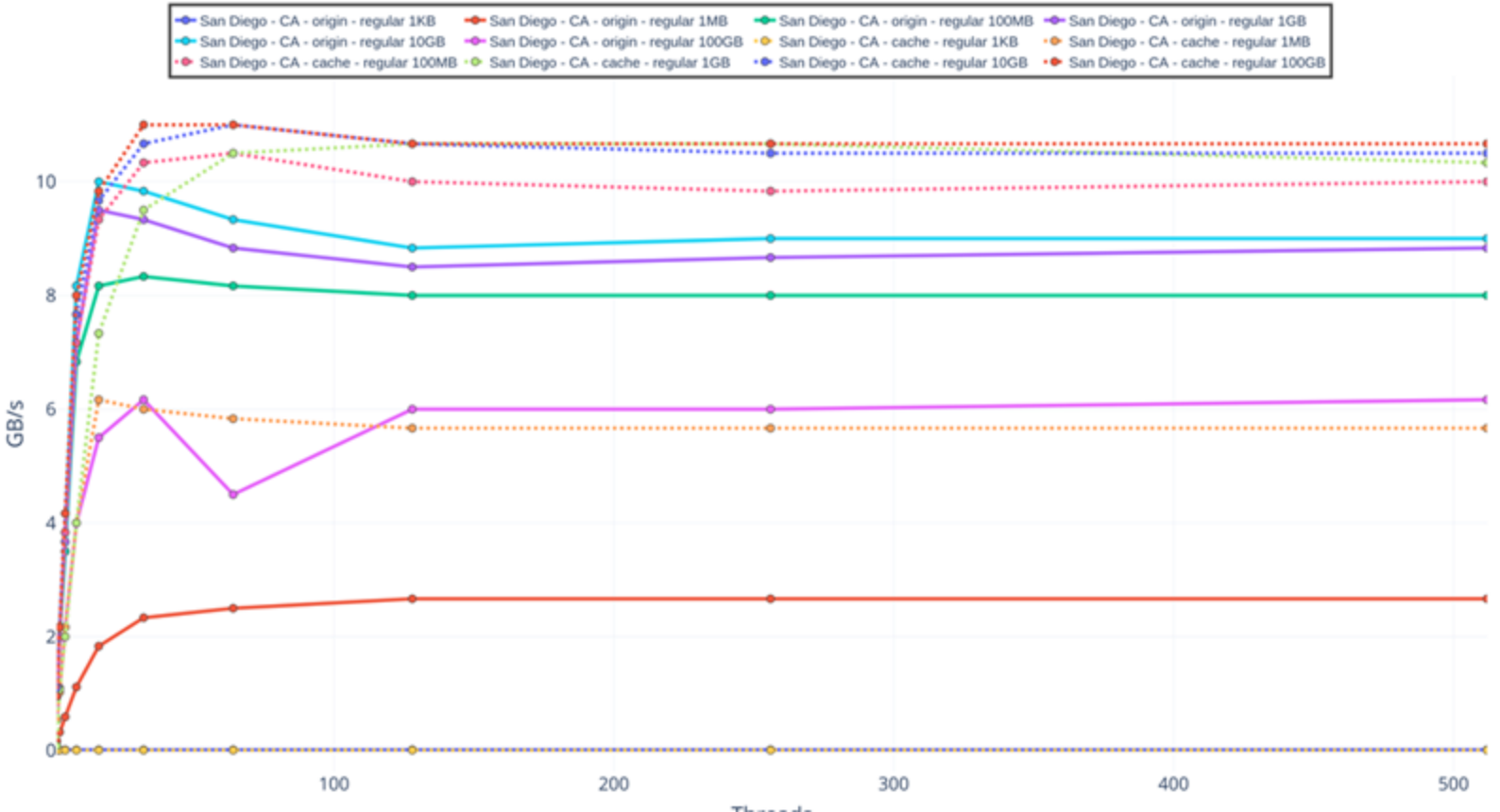
Impact: 4

Description. Test the transfer rate using six file sizes (1KB, 1MB, 100MB, 1GB, 10GB, 100GB) using 1, 2, 4, and 8 streams and EL7 vs EL9 as K8S OS POD forcing the redirector be used in each transfer.

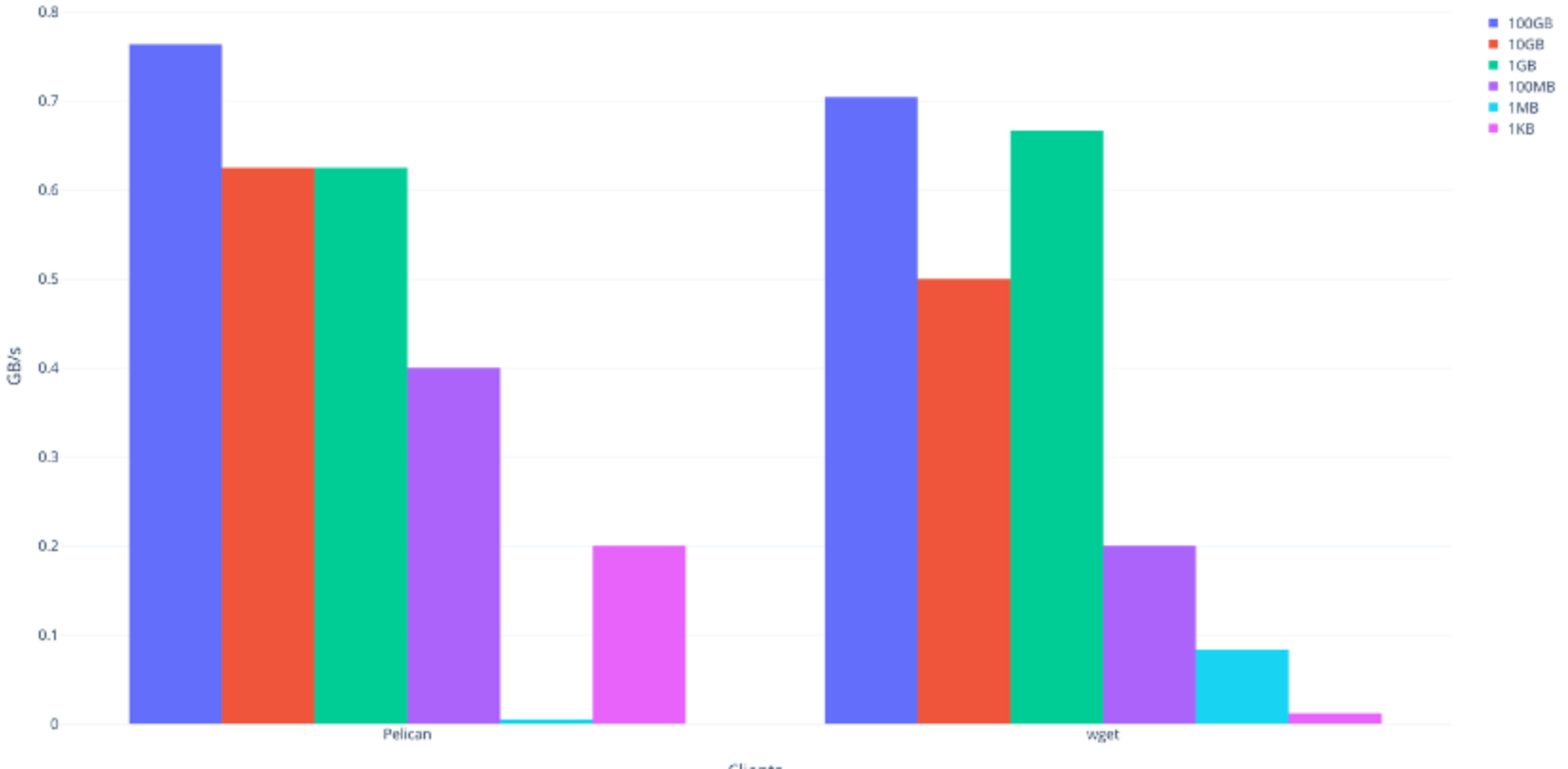

Refresh dashboard
Received
150 Gb/s
125 Gb/s
100 Gb/s
75 Gb/s
50 Gb/s
25 Gb/s
0 b/s
17:23
17:24
17:25
17:26
17:27
17:28
17:29
17:30
max
avg
current
vlan.4071 128 Gb/s 116 Gb/s 116 Gb/s
vlan.4072 22.7 Mb/s 5.85 Mb/s 1.11 kb/s
vlan.4073 20.2 Mb/s 2.93 Mb/s 1.10 kb/s
vlan.4074 15.9 Mb/s 3.59 Mb/s 6.80 Mb/s

	Origin	Cache
San Diego	0.088/0.109/0.186/0.038 ms 0 km	0.066/0.178/0.339/0.092 ms 0 km
Chicago	47.331/47.350/47.391/0.023 ms 2,784.10 km	47.337/47.353/47.394/0.021 ms 2,784.10 km
Jacksonville	51.324/56.352/57.381/0.023 ms 3,359.86 km	51.325/56.354/57.383/0.023 ms 3,359.86 km

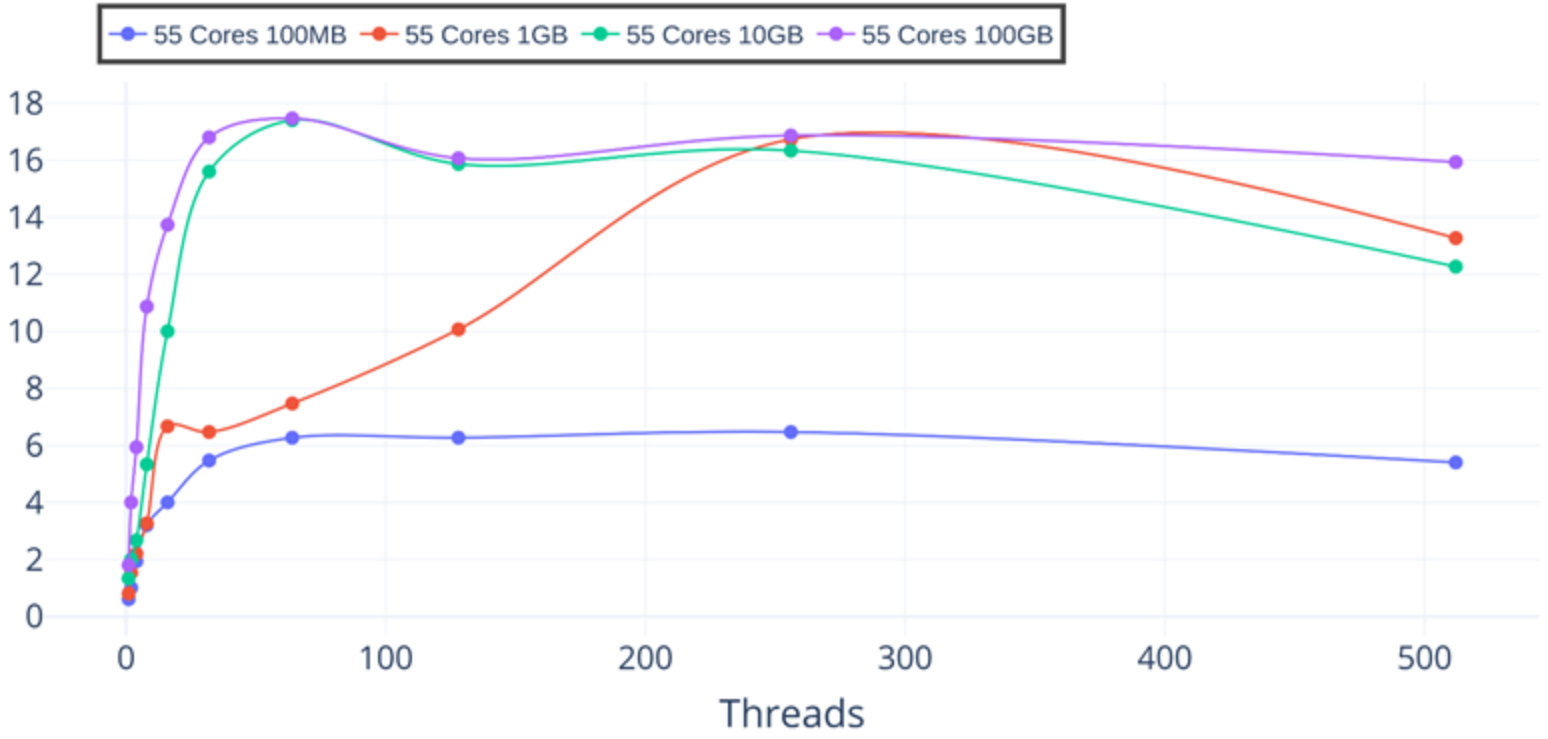
San Diego to San Diego



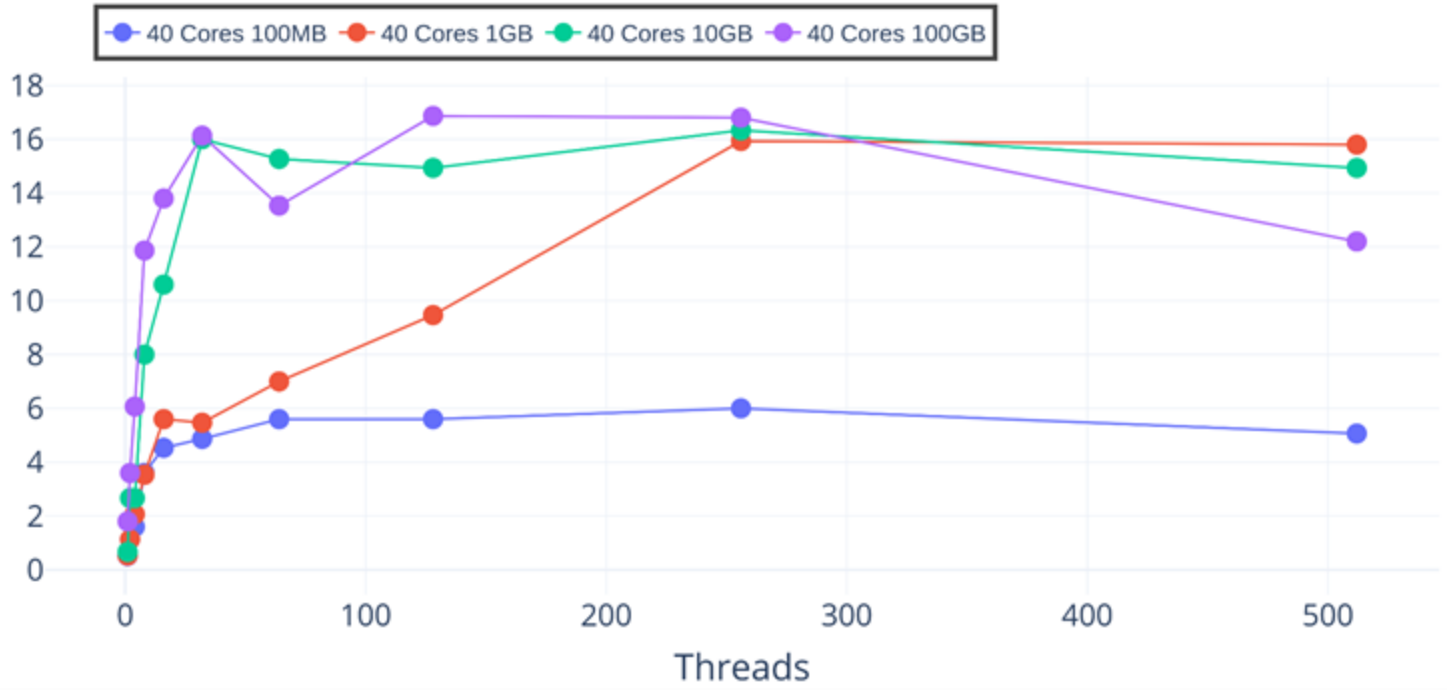
Clients


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
GB/s
100GB
10GB
1GB
100MB
1MB
1KB
Pelican
wget

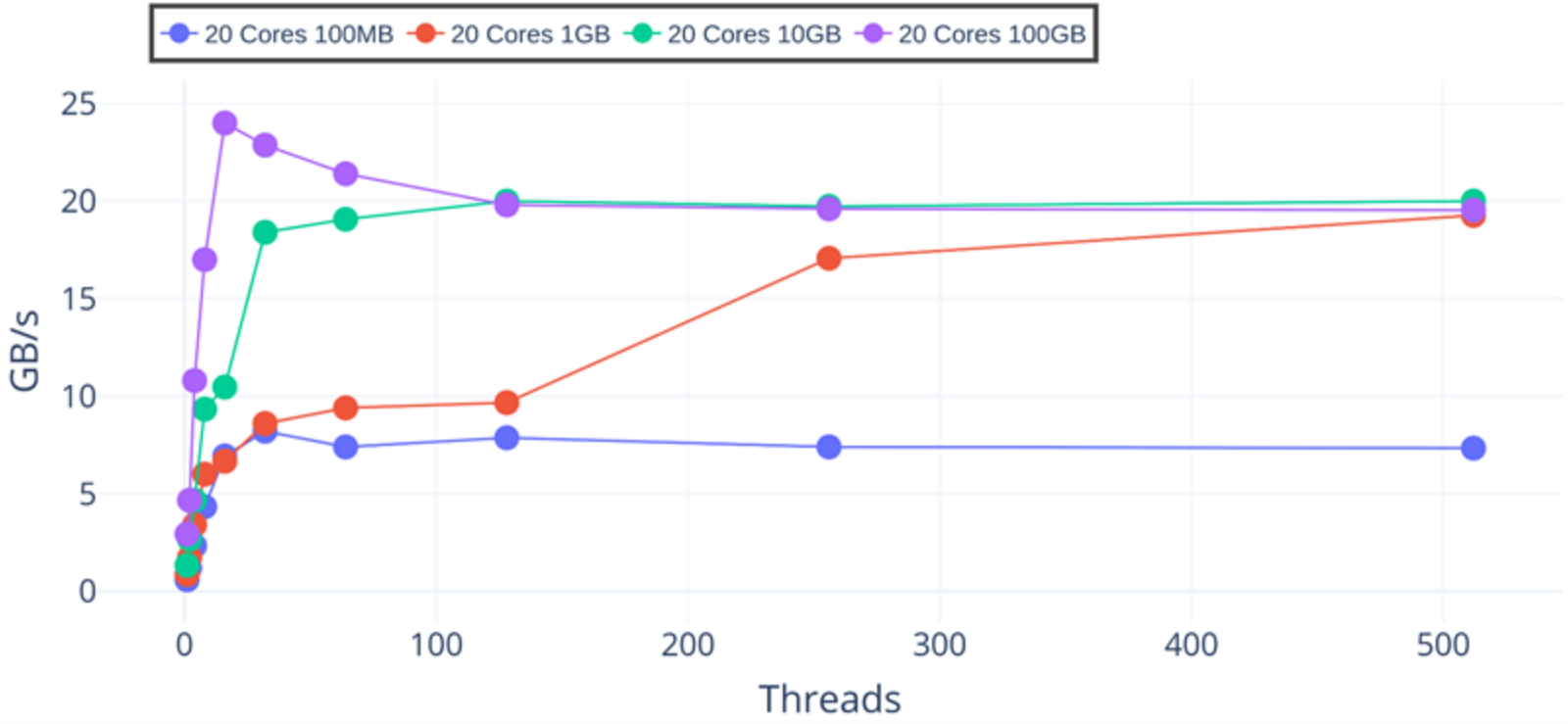
Using 55 cores from 60



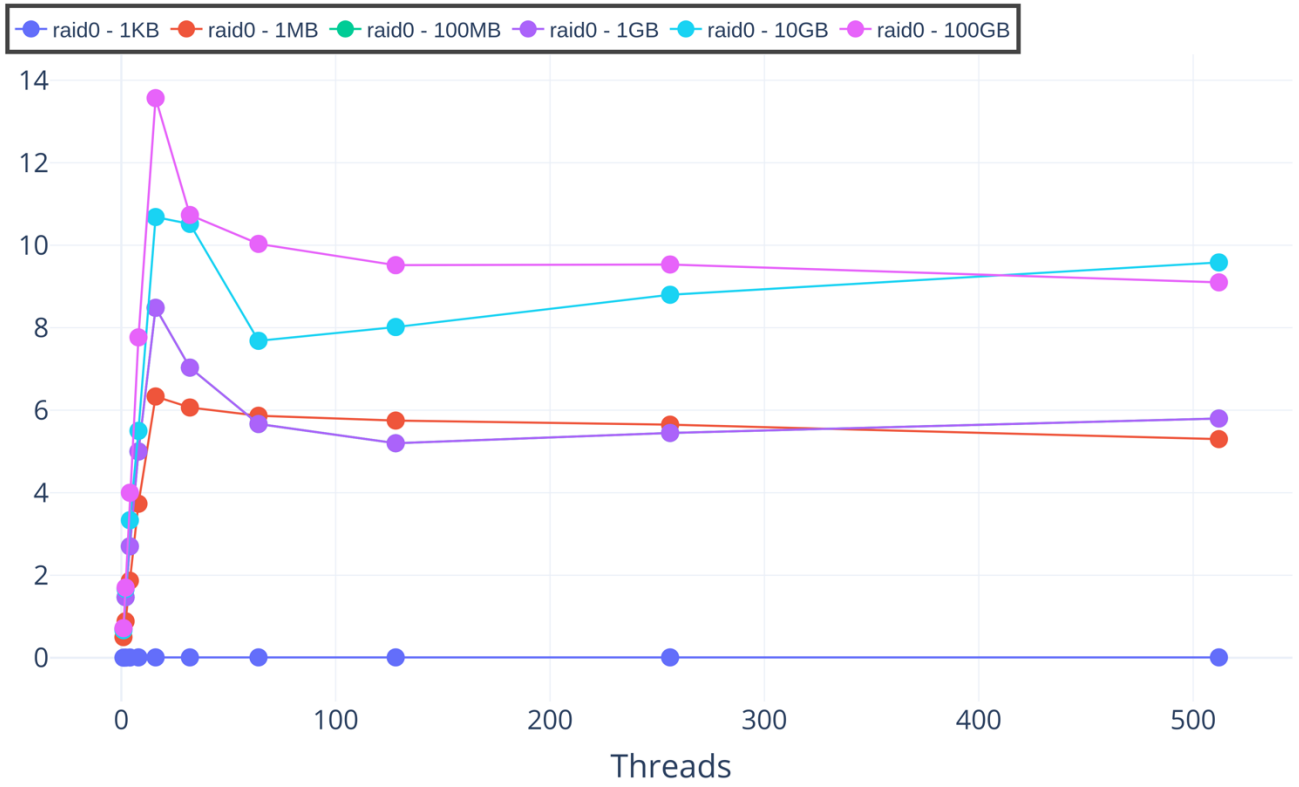
Using 40 cores from 60



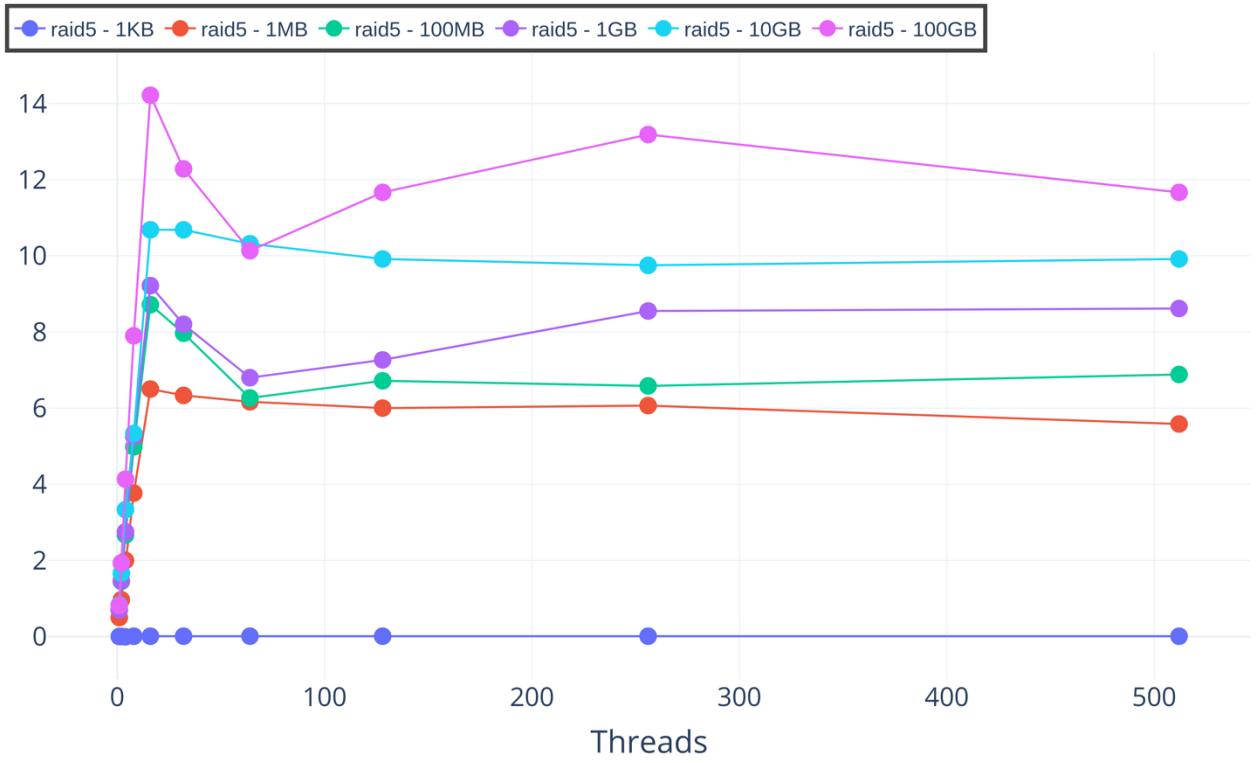
Using 20 cores from 60



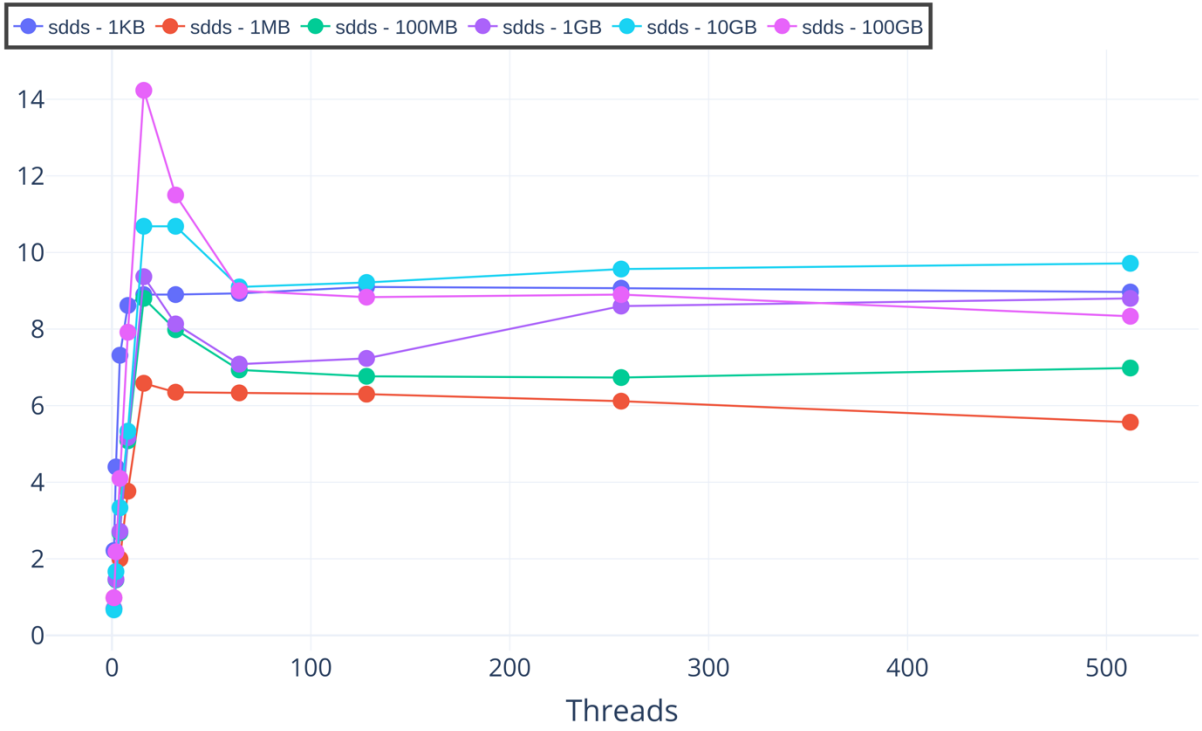
Raid0


raid0 - 1KB
raid0 - 1MB
raid0 - 100MB
raid0 - 1GB
raid0 - 10GB
raid0 - 100GB
14
12
10
8
6
4
2
0
0
100
200
300
400
500
Threads

Raid5


raid5 - 1KB
raid5 - 1MB
raid5 - 100MB
raid5 - 1GB
raid5 - 10GB
raid5 - 100GB
14
12
10
8
6
4
2
0
0
100
200
300
400
500
Threads

SSD


sdds - 1KB
sdds - 1MB
sdds - 100MB
sdds - 1GB
sdds - 10GB
sdds - 100GB
14
12
10
8
6
4
2
0
0
100
200
300
400
500
Threads

XrootD S3 plugin

- It is able to connect to S3 and OSN
- The plugin works, however it is close to 8 times slower than accessing directly the access point.

- https://sdsc-s3-origin.nationalresearchplatform.org/osn-sdsc/1g:
  - File size: 1000 MB.
  - Download speed: ~5.71 MB/s.
  - Time taken: 3 minutes.

XrootD S3 plugin

- https://rice1.osn.mghpcc.org/osn-sdsc/1g:
  - File size: 1000 MB.
  - Download speed: 290 MB/s.
  - Time taken: ~5 seconds (much faster than the first).
- https://osdf-director.osg-htc.org/osn-sdsc/1g:
  - Redirected to fdp-d3d-cache.nationalresearchplatform.org.
  - File size: 1000 MB.
  - First download speed: 1.99 MB/s, taking about 9 minutes (slower).
  - Second download speed: 407 MB/s, taking about 2.5 seconds (much faster).

Acknowledgements

- This work was partially supported by the NSF grants OAC-2112167, OAC-2030508, OAC-1841530, OAC-1836650, the CC* program, and in kind contributions by many institutions including ESnet, Internet2, and the Great Plains Network.