Open Science Grid

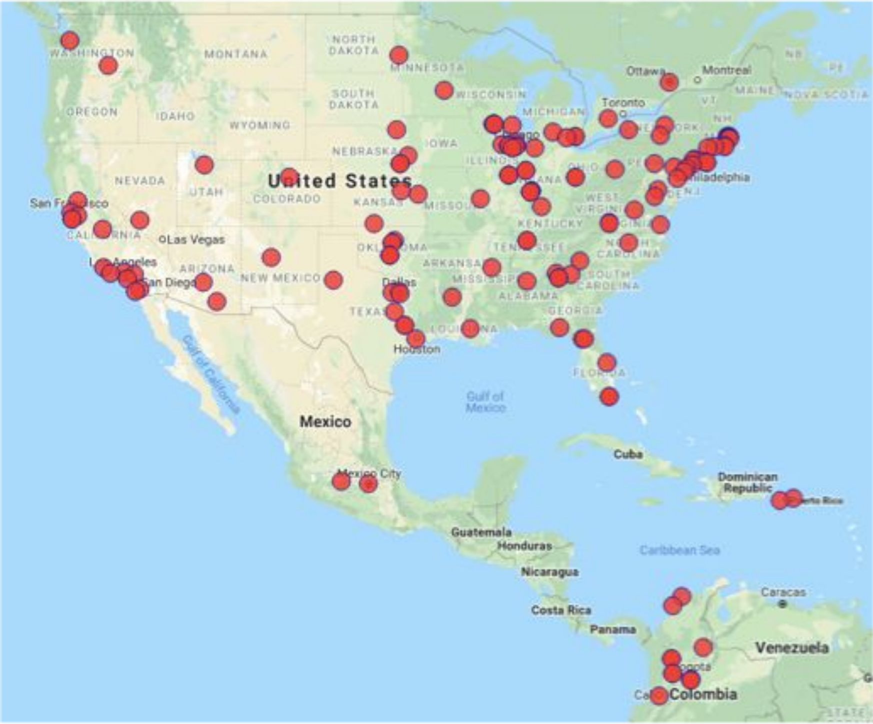
Creating a content delivery network for general science on the backbone of the Internet using XCache(s).

Edgar Fajardo

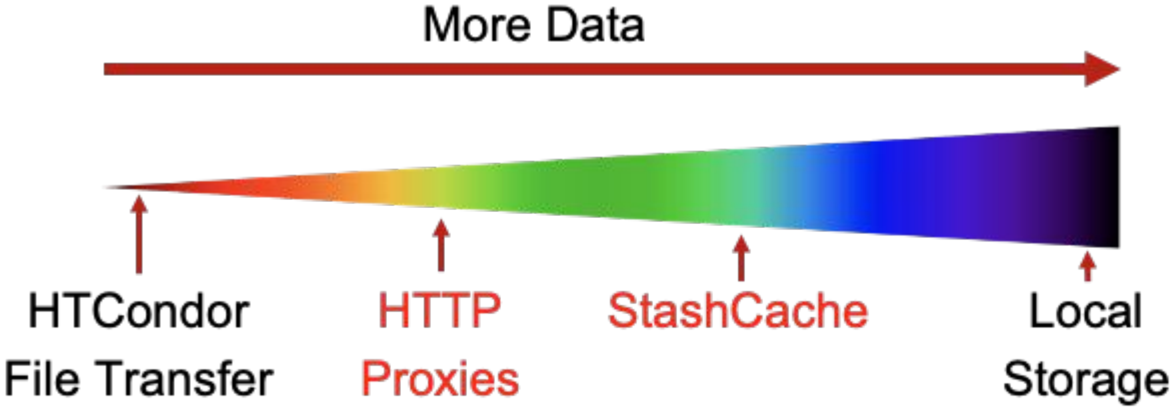
In Collaboration with: Brian Lin, John Hicks, Marian Zvada, Derek Weitzel, Mat Selmeci, Pascal Paschos

Introduction to Open Science Grid (OSG)

- OSG aggregates compute resources from over 100 campuses both nationally and internationally
- OSG also serves almost 40 different user communities, each with its own set of data origins
- With a handful having really large input datasets
- Networking is essential to deliver data from origins to compute endpoints



When is StashCache useful?



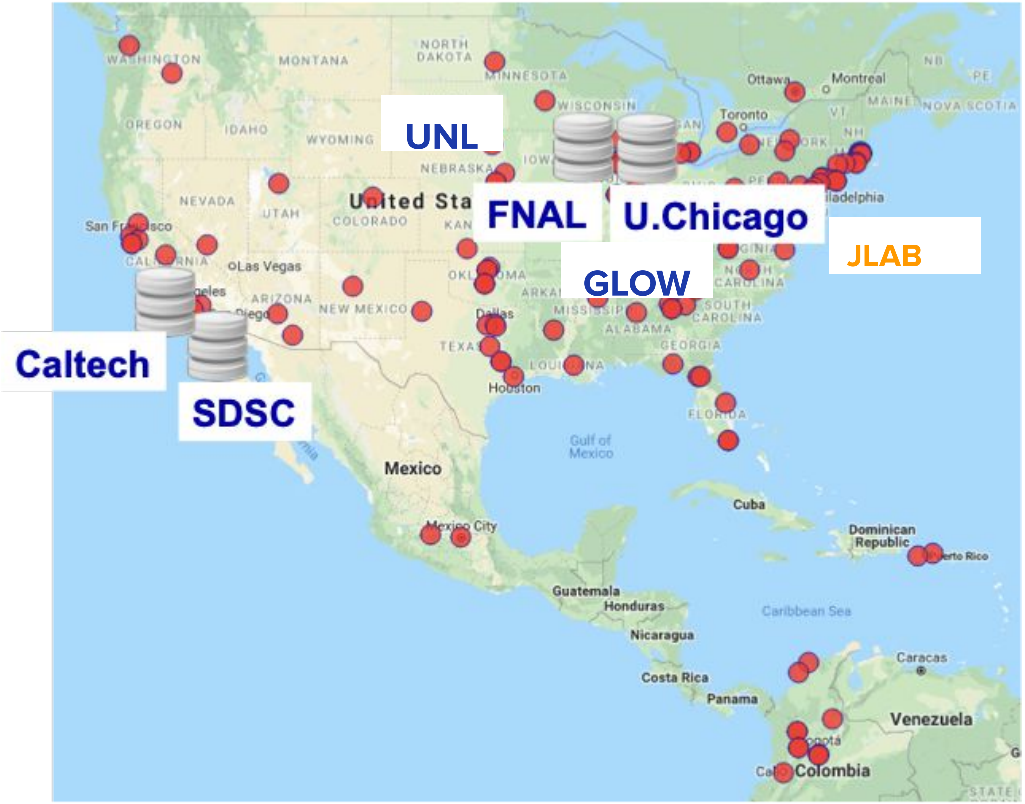
- Credit: OSG User School

When is StashCache useful?

Data Size (per job)	Method of Delivery
words	within executable or arguments?
tiny – 100MB per file	HTCondor file transfer (up to 1GB total)
100MB – 1GB, shared	download from web server (local caching)
1GB - 20GB, unique or shared	StashCache (regional replication)
20 GB - TBs	shared file system (local copy, local execute servers)

- Credit: OSG User School

OSG Data Origins



- OSG supports different scientific communities all across the science spectrum.
- These communities happen to have a “Golden copy” of their data (data origin) all around the country

FNAL: Fermilab based HEP Experiments

U.Chicago: General OSG Community

Caltech: Public LIGO Data Releases

SDSC: Simons Foundation

UNL: LIGO Data Release

JLAB coming next. CLAS12, GLUEX, EIC

Implications of this model

- Data is moved from its origin to the jobs using the network.
- If a data file is reused by several jobs the same file travels the network several times. For example:
    - LIGO - time shifter analysis
    - Biology-related communities – DNA matching
    - Any kind of parameter estimation over the same data set

Hence: Caching in the network

Benefits of data caching in the network

- Reduce origin to backbone data transfers:
  - Data only travels once from the origin to the cache
  - Reduces stress on data origins
  - Increases redundancy
- Increase CPU efficiency for latency sensitive applications
  - Less time wasted waiting for data
- Benefits both types of applications
  - Lower RTT greatly benefits latency sensitive applications
  - Reduced data origin server congestion allows for higher endpoint bandwidth

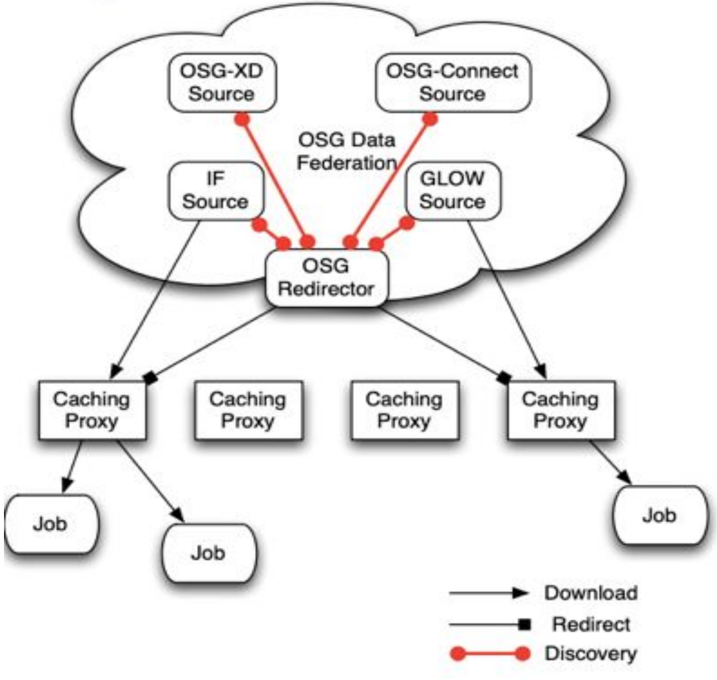
OSG Caching Solution: StashCache

Introduction to Stashcache

- Caching infrastructure based on SLAC XRootD server & XRootD protocol.
- Cache servers are placed at several strategic cache locations across the OSG.
- Jobs utilize GeoIP to determine the nearest cache
- Job talks to the cache using HTTP(S) via CVMFS

Powered by: XRootD

Image taken from Brian's slides



Implications for the Infrastructure

- An organization can join the federation with their own “data origin” and their own partition of the global namespace. Like /gwosgc, /osgconnect, /pnfs/fnal/…/dune
- A cache owner can decide on caching policies for different parts of the namespace.
- This allows the owner to selectively serve only a subset of the community that uses the federation.

Stashcache from user's perspective

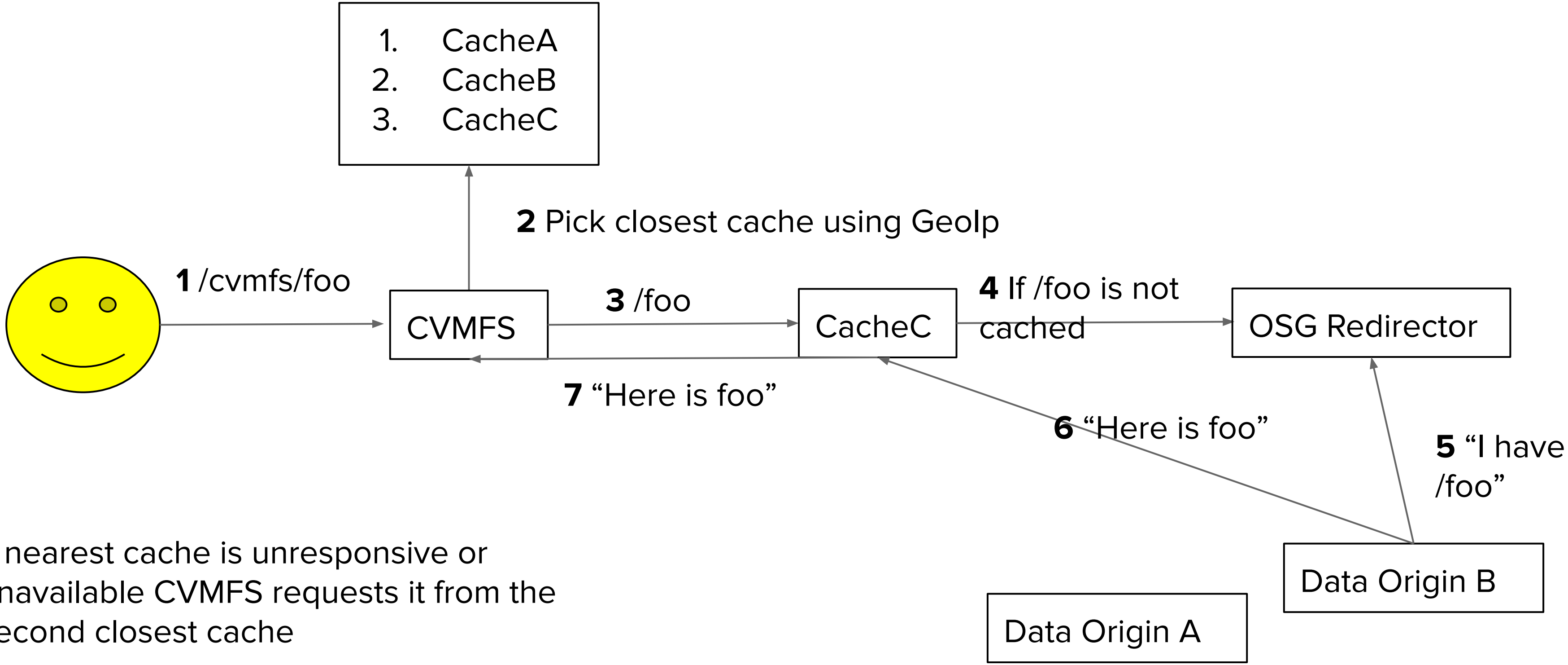
- User jobs access their data either by POSIX mount /cvmfs/foo
- Or via an OSG tool called stashcp:

```
$ stashcp /osgconnect/public/<username>/blast.db blast.db
```

  - Under the hood stashcp tries to obtain the files via CVMFS if available.
  - stashcp will use the same GeoIP to location caches
  - Not all namespaces are available via CVMFS (Data owners have to request it)
  - stashcp provides an instantaneous view of the namespace, CVMFS is delayed ~1-8 hours.

Stashcache should be invisible from the user's perspective

StashCache behind the scenes



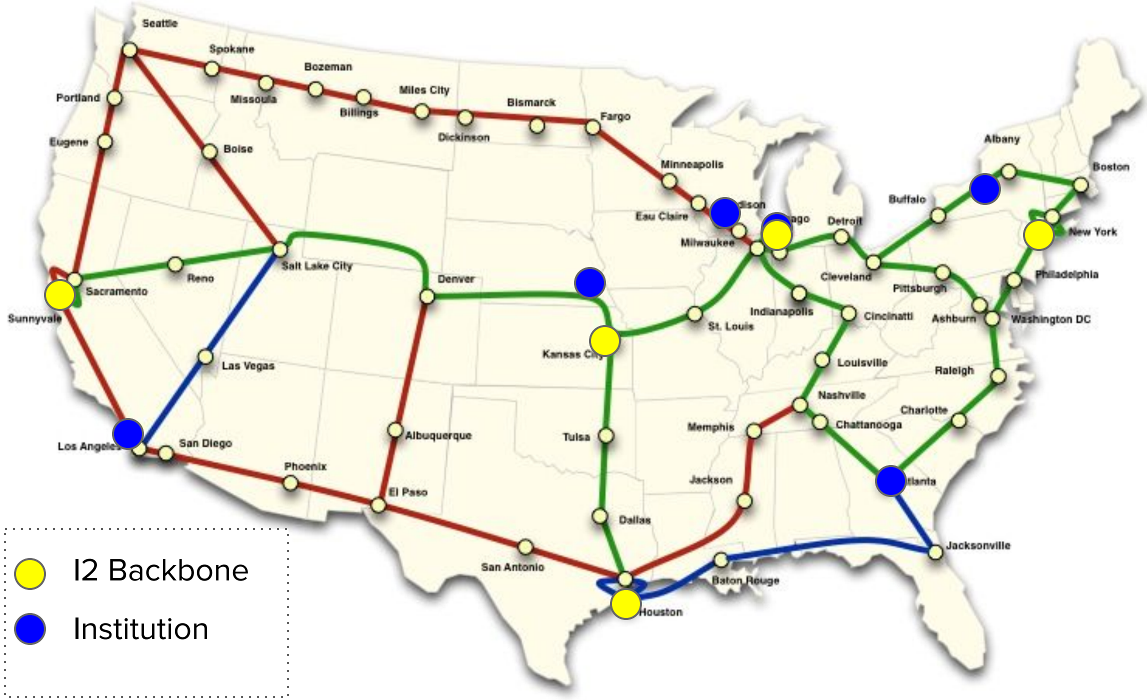
If nearest cache is unresponsive or unavailable CVMFS requests it from the second closest cache

Stashcache should be invisible from the user's perspective

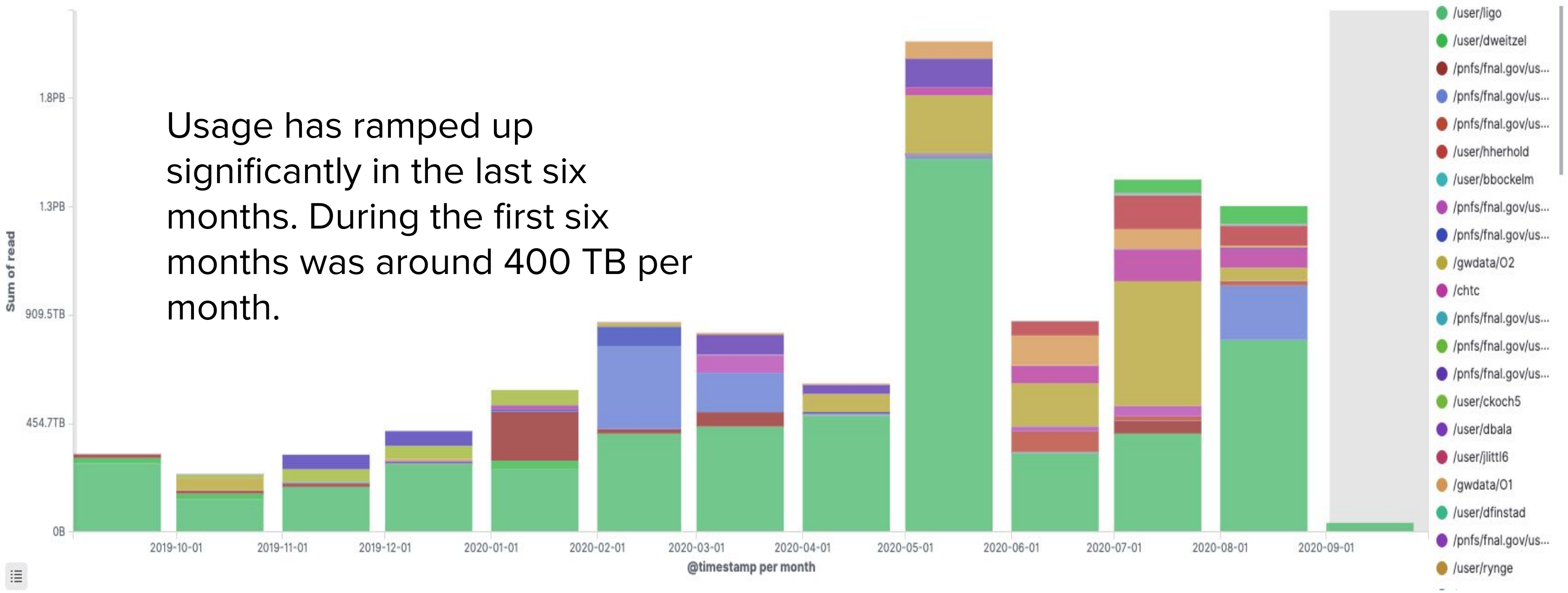
Caches in the backbone

- A joint project between Internet2 and OSG to place several caches on the backbone of the Internet2
- Originally three caches were deployed in the backbone: KC, Chicago and Manhattan.
- Since OSG is moving to a DevOps model all the new caches were deployed using Kubernetes for maximum flexibility of deployment (i.e one day these are caches tomorrow someone can deploy another container for bandwidth testing).
- This gave rise in 2020 to the following cache topology.

StashCache Locations (US)

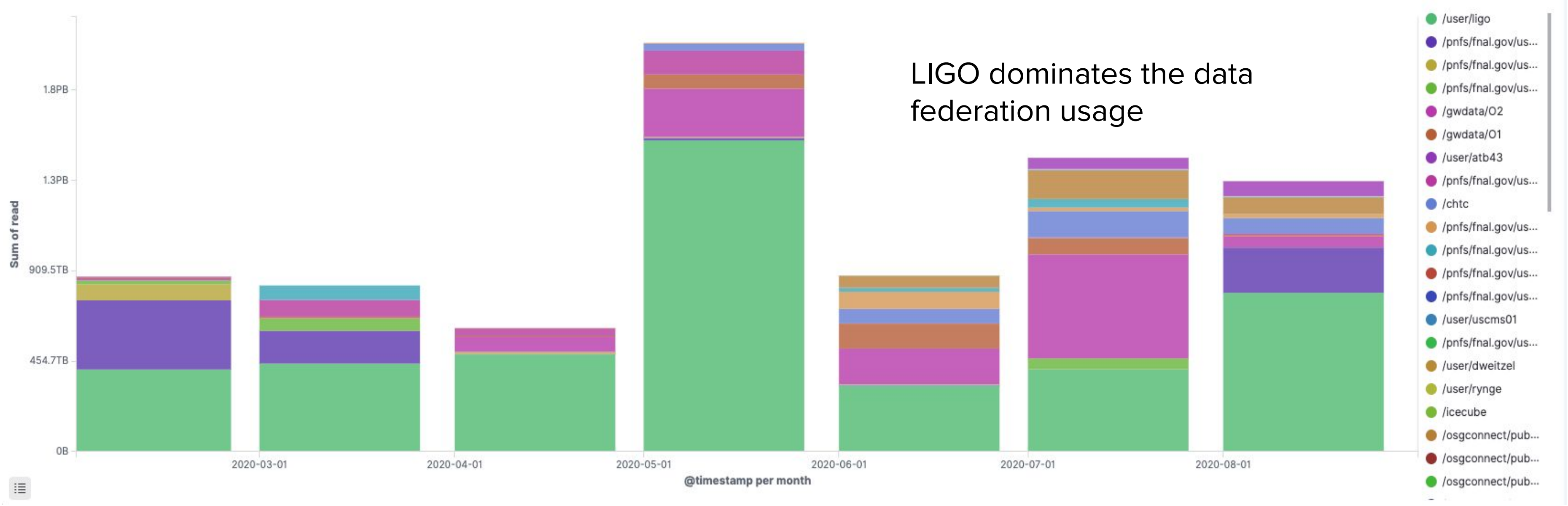


Usage in the last year



Usage has ramped up significantly in the last six months. During the first six months was around 400 TB per month.

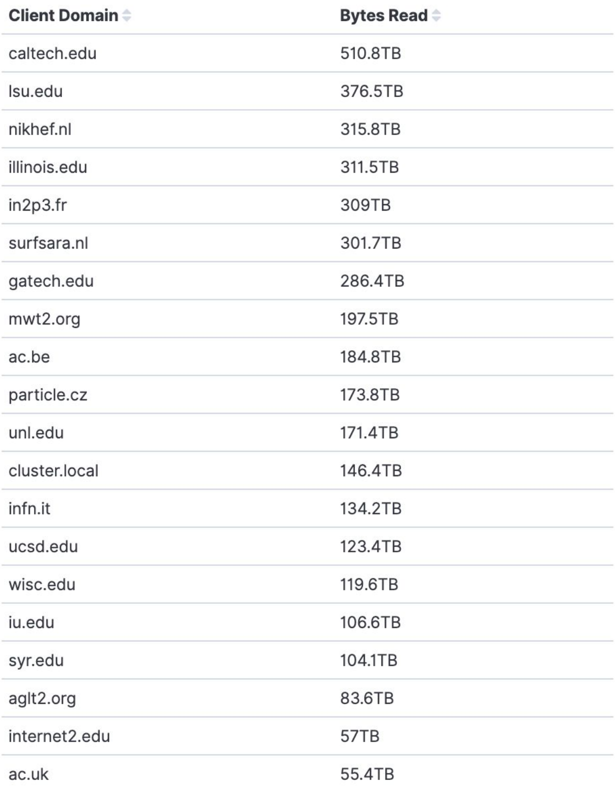
Last six months of usage



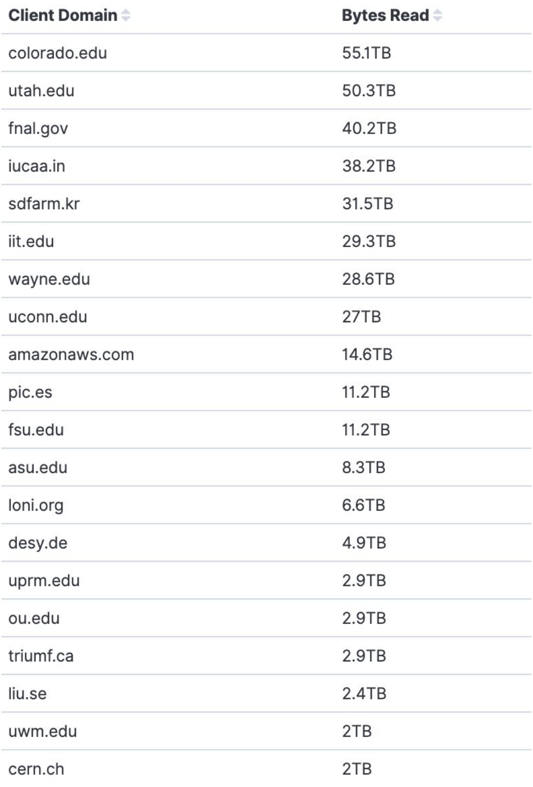
LIGO dominates the data federation usage

This is an example of the last six months of data delivery by the content delivery network for a an average of 1.5 PB of data delivered per month.

Where was this data read from (40+ sites)



Client Domain	Bytes Read
caltech.edu	510.8TB
lsu.edu	376.5TB
nikhef.nl	315.8TB
illinois.edu	311.5TB
in2p3.fr	309TB
surfsara.nl	301.7TB
gatech.edu	286.4TB
mwt2.org	197.5TB
ac.be	184.8TB
particle.cz	173.8TB
unl.edu	171.4TB
cluster.local	146.4TB
infn.it	134.2TB
ucsd.edu	123.4TB
wisc.edu	119.6TB
iu.edu	106.6TB
syr.edu	104.1TB
aglt2.org	83.6TB
internet2.edu	57TB
ac.uk	55.4TB



Client Domain	Bytes Read
colorado.edu	55.1TB
utah.edu	50.3TB
fnal.gov	40.2TB
iucaa.in	38.2TB
sdfarm.kr	31.5TB
iit.edu	29.3TB
wayne.edu	28.6TB
uconn.edu	27TB
amazonaws.com	14.6TB
pic.es	11.2TB
fsu.edu	11.2TB
asu.edu	8.3TB
loni.org	6.6TB
desy.de	4.9TB
uprm.edu	2.9TB
ou.edu	2.9TB
triumf.ca	2.9TB
liu.se	2.4TB
uwm.edu	2TB
cern.ch	2TB

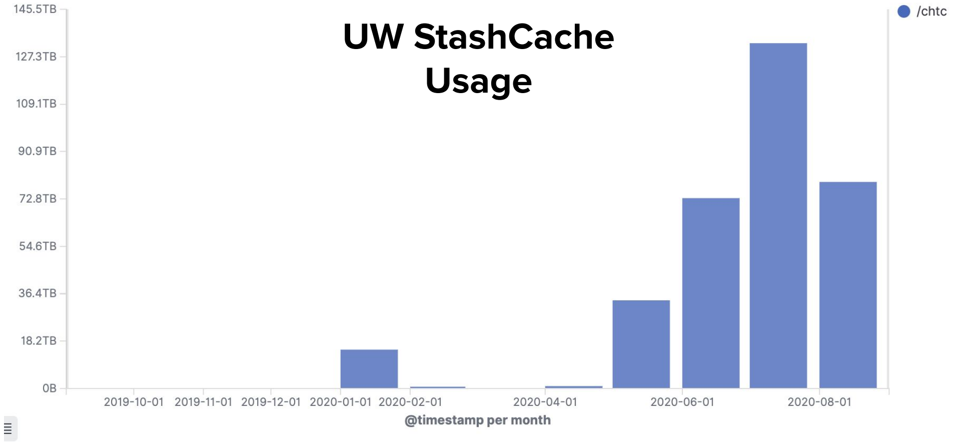
Cache Usage

Collaboration	Working Set	Data Read	Reread Multiplier	
DUNE	25GB	131TB	5.4k	FNAL
LIGO (private)	41.4TB	3.8PB	95	LIGO
LIGO (public)	4.3TB	1.5PB	318	
MINERVA	351GB	116TB	340	FNAL
DES	268GB	17TB	66	
NOVA	268GB	308TB	1.2k	
odgerk	67GB	541TB	8.3k	Single PI

Data pulled from federation in 6 month period 3-8/2020

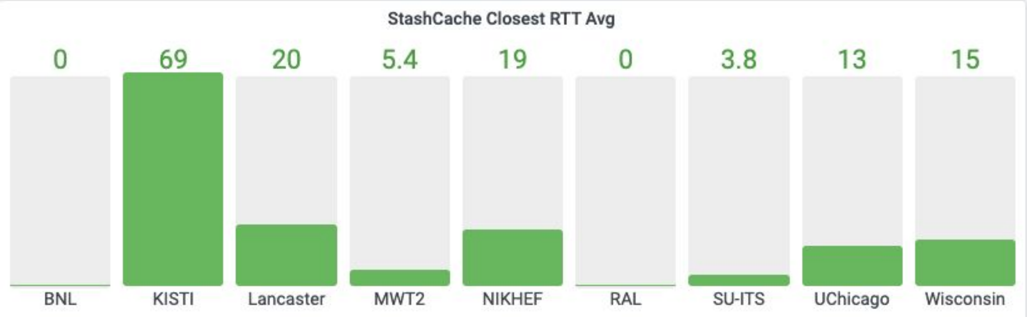
Growing Cache Usage

- 4 Single PIs downloading more than 1TB in the last 6 months
  - Previous 6 months was 2
- 2 campuses significantly increased their usage (UNL and UW)
  - UW downloaded 312 TB in the last 6 months, compared to 12TB the previous 6.
  - UNL downloaded 131 TB in the last 6, compared to 0 the previous 6.

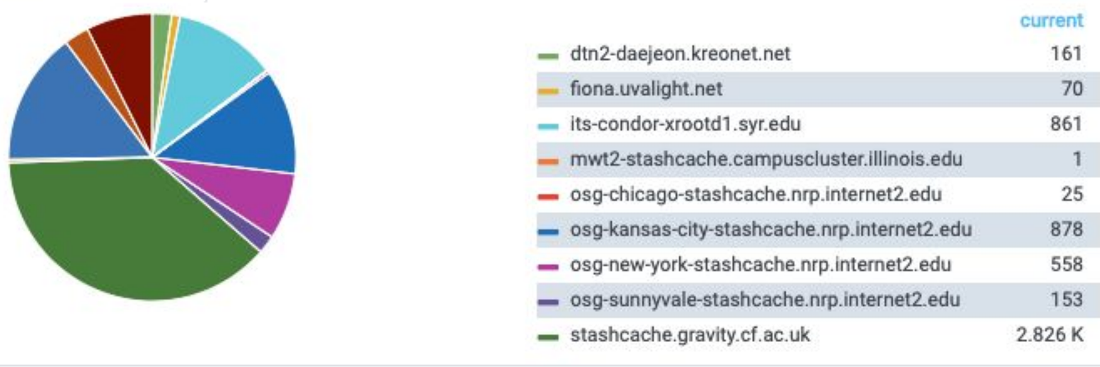


Pilot Integration

StashCache Closest RTT Avg



StashCache Closest



	current
dtn2-daejeon.kreonet.net	161
fiona.uvalight.net	70
its-condor-xrootd1.syr.edu	861
mwt2-stashcache.campuscluster.illinois.edu	1
osg-chicago-stashcache.nrp.internet2.edu	25
osg-kansas-city-stashcache.nrp.internet2.edu	878
osg-new-york-stashcache.nrp.internet2.edu	558
osg-sunnyvale-stashcache.nrp.internet2.edu	153
stashcache.gravity.cf.ac.uk	2.826 K

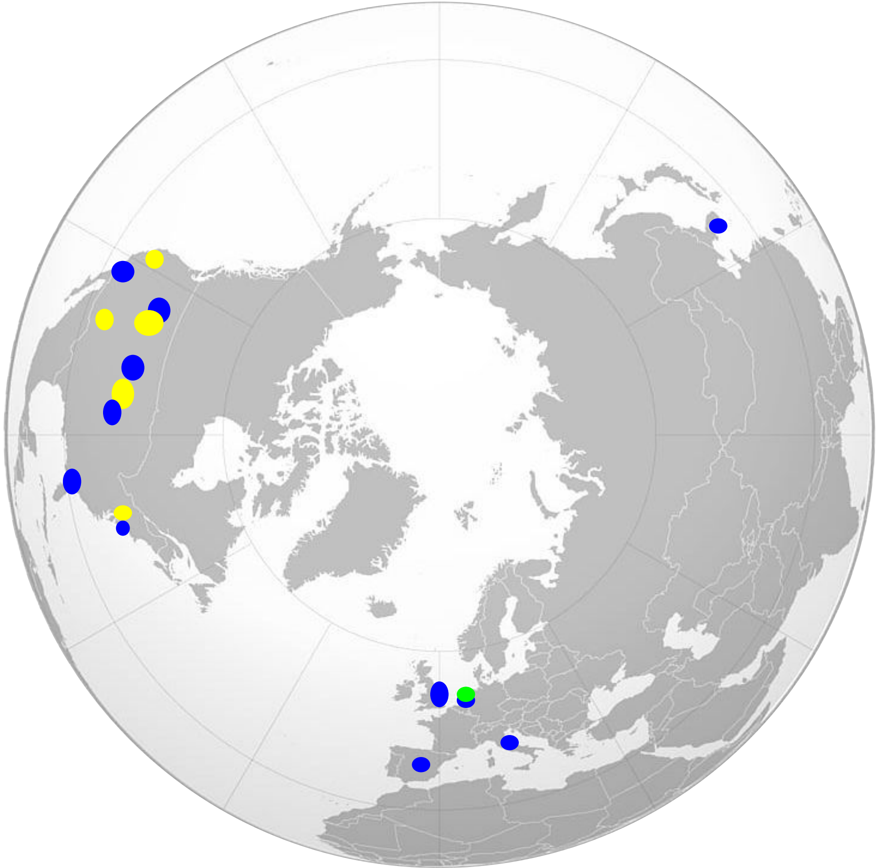
Each LIGO pilot pushes a ClassAd of where its closest cache is and what is the latency. The results can be seen in here. Use (guest/guest)

StashCache Location (WorldWide)

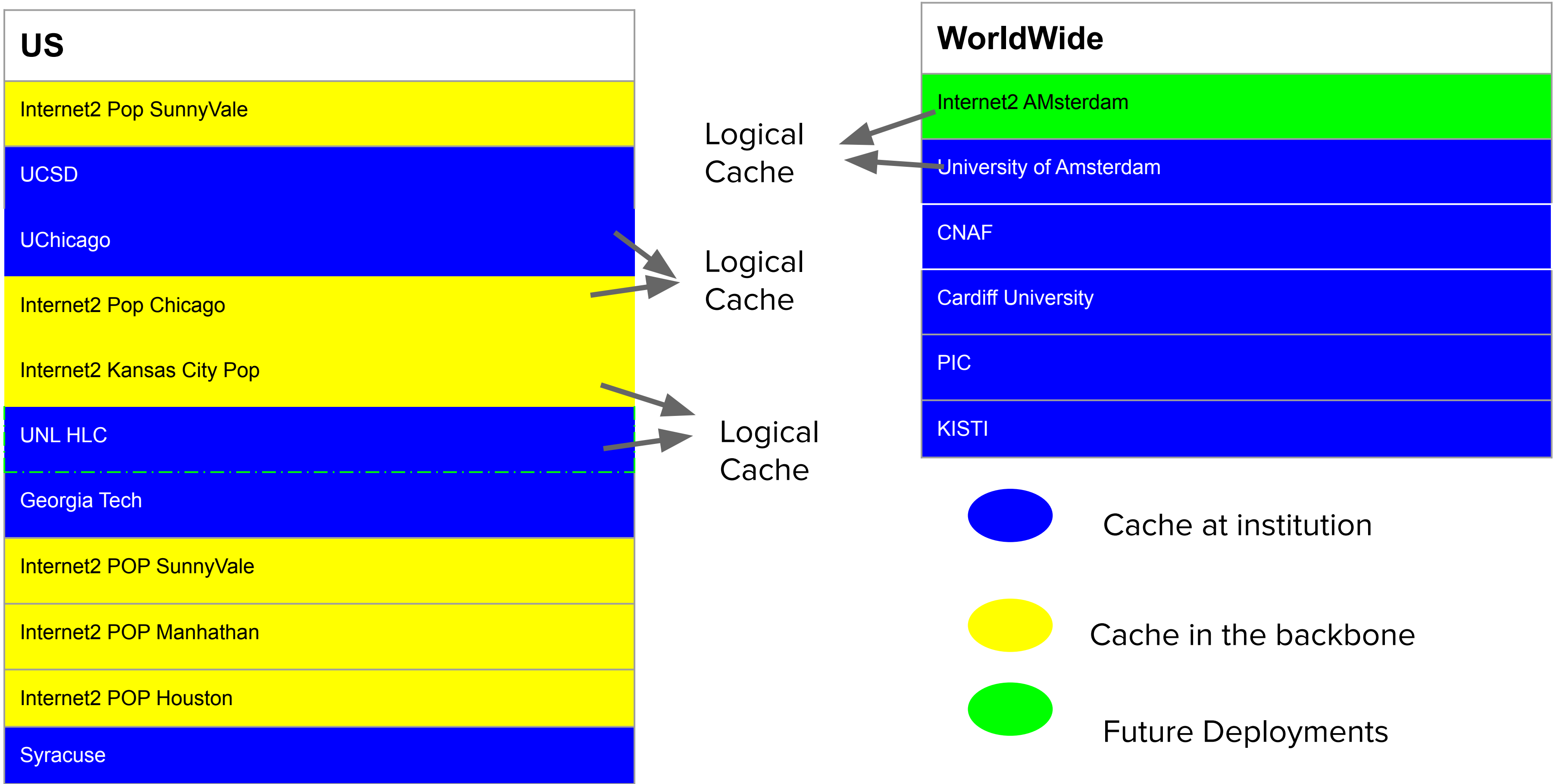
● Cache at institution

● Cache in the backbone

● Future Deployments



StashCache list



US
🟡 Internet2 Pop SunnyVale
🔵 UCSD
🔵 UChicago
🟡 Internet2 Pop Chicago
🟡 Internet2 Kansas City Pop
🔵 UNL HLC
🔵 Georgia Tech
🟡 Internet2 POP SunnyVale
🟡 Internet2 POP Manhathan
🟡 Internet2 POP Houston
🔵 Syracuse

WorldWide
🟢 Internet2 AMsterdam
🔵 University of Amsterdam
🔵 CNAF
🔵 Cardiff University
🔵 PIC
🔵 KISTI

🔵 Cache at institution

🟡 Cache in the backbone

🟢 Future Deployments

We want (to distribute) your data

- We are happy to help you distribute it using our cache network:
- You only need to install an XRootD Origin in top of the file system that holds your data.
- We support several installations:
    - Kubernetes setup controlled by us
    - Docker installed managed by you with our docker image
    - Bare metal (RPM) install in which you do all the work.
- What you need:
    - A host with (at least) a 10Gbps network connection to the WAN.

Want to host a cache at your institution?

- As with the origins we are happy to help you run a cache at your institution to advance science.
- Specially if your institution is relatively far from the any of the dots showed in the slides before.
- Please contact us

What can we learn from Industry?

- CDNs such as Cloudflare, Akamai, and Google have pioneered this space, lets learn from them!
- Kubernetized Operations
- Authenticated Access (SciTokens + TLS)
- Nearest Cache Choice
- Better universal clients (stashcp replacement)

Conclusions

- We built a data delivery network for general science purposes that profits from in the network caching
- Kubernetes was the building block to deploy this worldwide agile infrastructure.
- StashCache usage is growing, especially among individuals and campuses.
- Applying the OSG model: **Leverage large experiment’s technology investments to benefit small or single PI research teams**

Acknowledgments

- The authors would like to thank the funding agencies for this work, in particular the following grants:
  - PRP: NSF OAC-1541349
  - TNRP: NSF OAC-1826967
  - CESER: NSF OAC-1841530
  - OSG: NSF MPS-1148698
- Also we would like to thank Internet2 for providing the infrastructure on the backbone and their whole collaboration on this project.
- The different system administrators that have agreed to deploy hardware at the computing centers:
  - GaTech, UNL, UCSD, CNAF, U of Amsterdam, UChicago, Syracuse.
- To the PRP admins who let use use the Kubernetes federation to deploy the caches.
- The XrootD Development team for the continuous improvement of the XCache.
- OSG Software team for the XCache packaging and StashCache containers.

Questions?