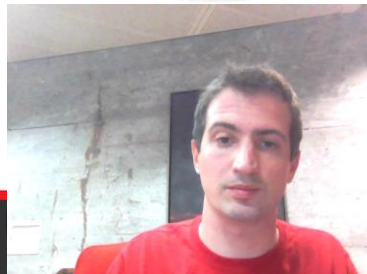
Jump's archive for the next decade

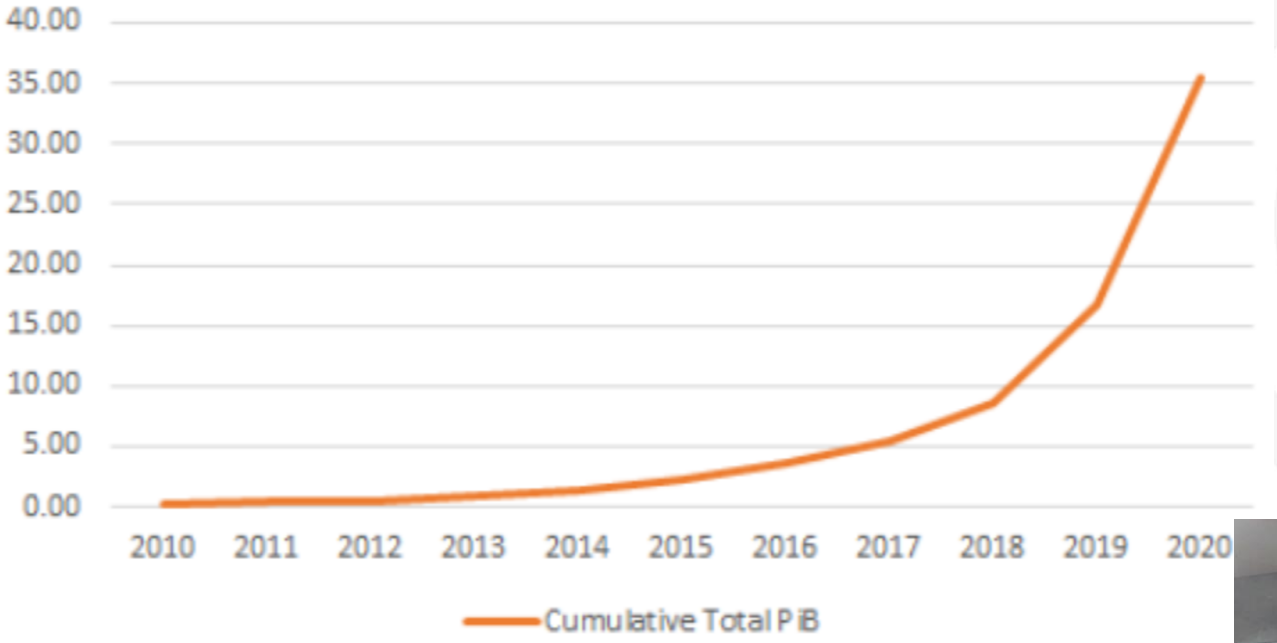
A high performance and scalable POSIX interface to an object store, via CernVM filesystem

jumptrading



HPC at Jump

Ten Years of Archive Growth (for one of several 'archives')



CVMFS for POSIX-like presentation

- CVMFS (CernVM-FS) is a FUSE filesystem developed at CERN by the computational physics community
- Used at Jump for years for internal software distribution
- Read path uses only outgoing HTTP requests
- Metadata stored in SQLite databases
- Data and metadata both easily cached
- Changes presented as filesystem revisions
  - Each revision is an internally consistent view of the archive
  - A host is always on “revision X” or “revision X+1”; never in between

A new design

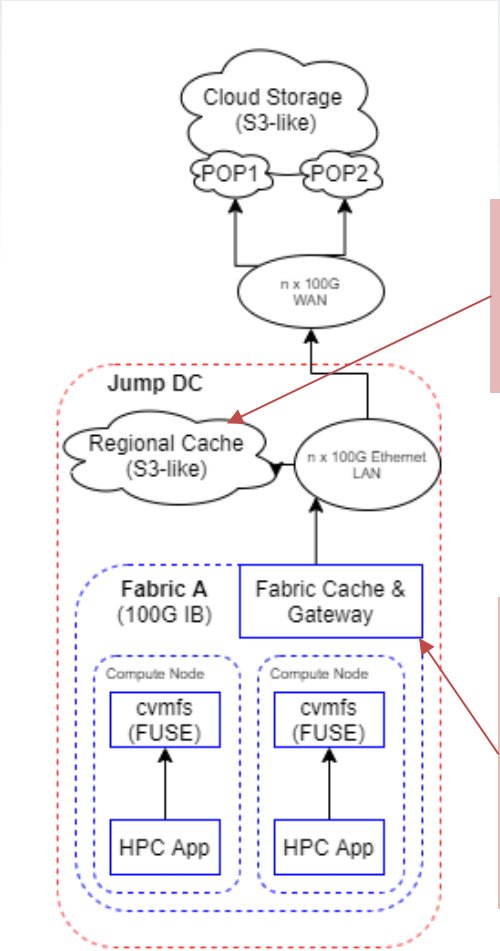
2xPOPs each with 2x100G

100G leaf/spine
Ethernet network

EDR (1 or 200G IB) fabric

CVMFS FUSE mount

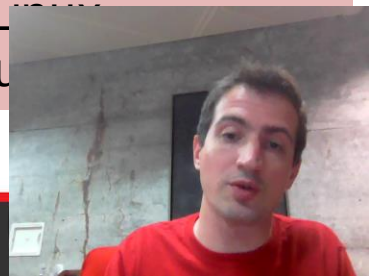
Unmodified application



Google Cloud Storage

~5PB VAST S3 Appliance
Mix of 3D XPoint (fast) and QLC (slow/economical)

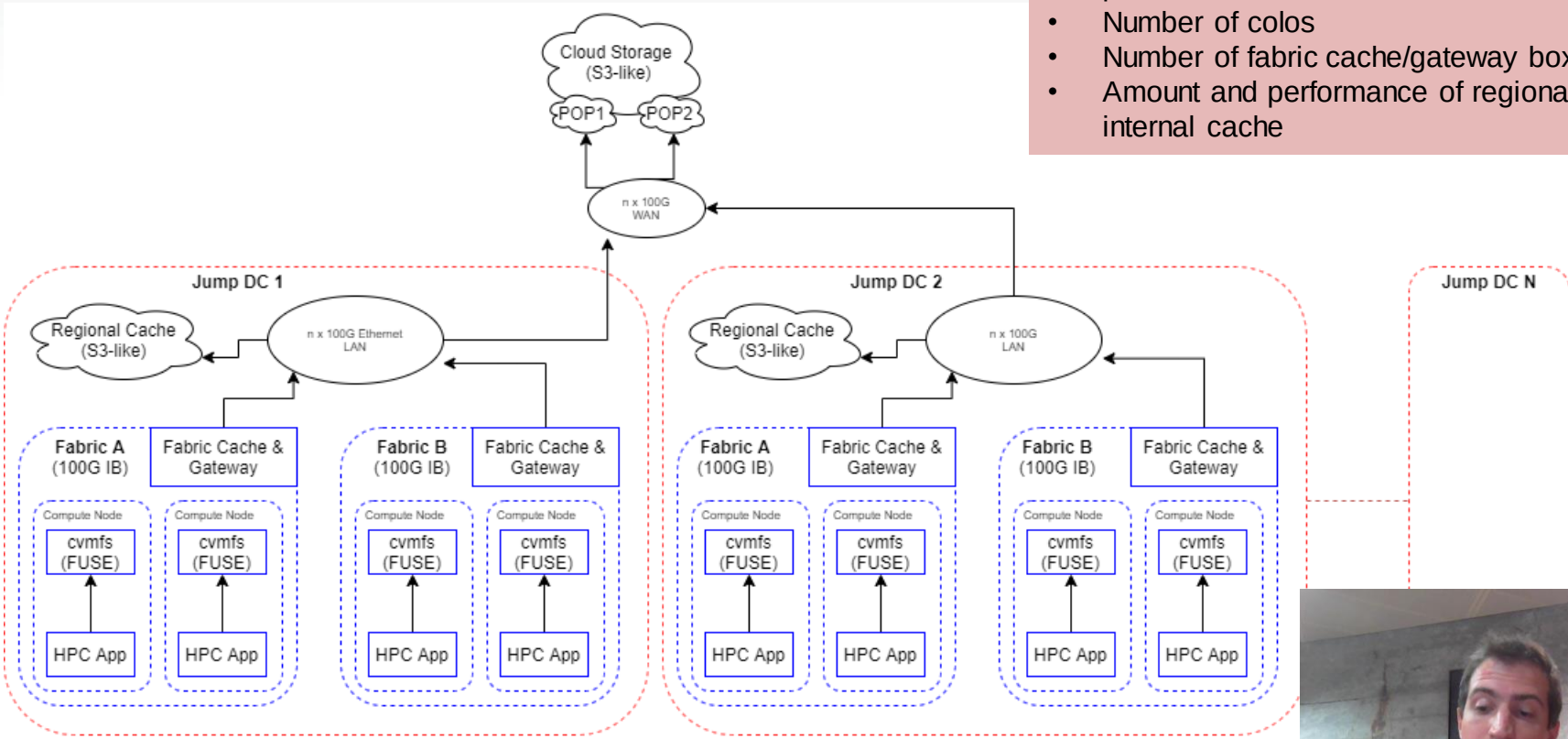
Cache/Gateway box
- Many NICs IB/Eth
- ~40T SSD
- Standard Li
- Varnish Plu

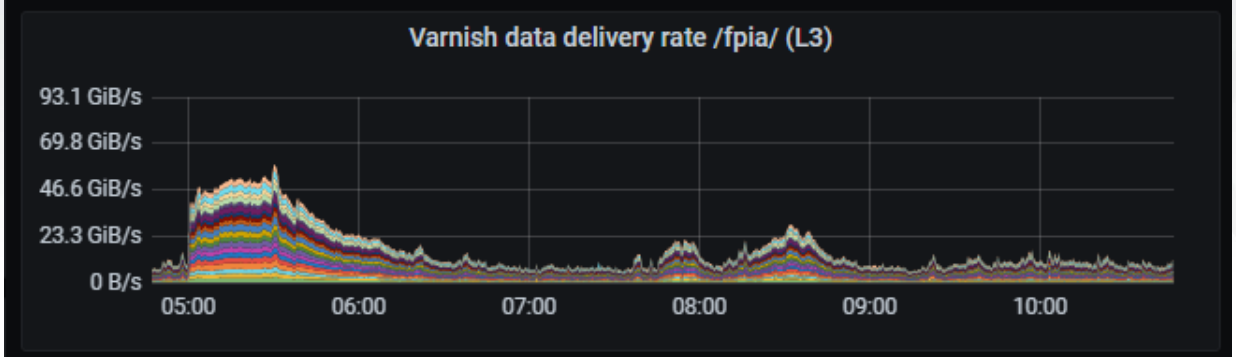


.. for the next 10 years

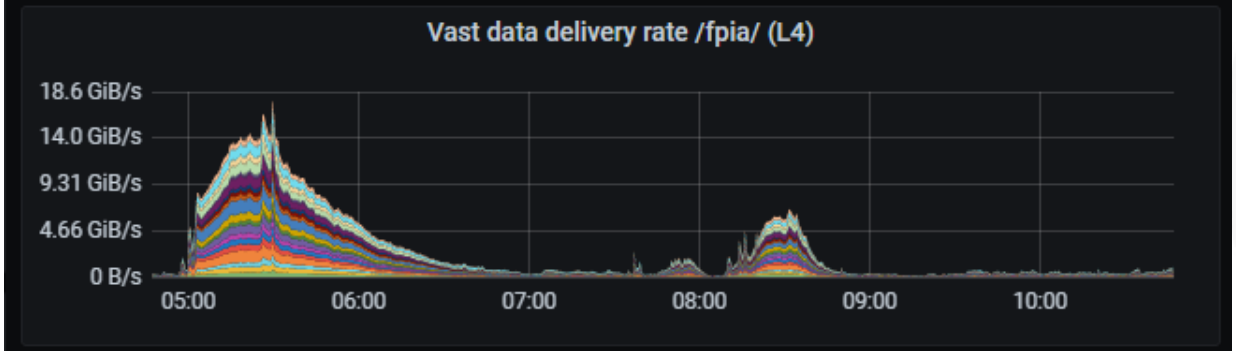
Can scale by orders of magnitude:

- Storage PB
- Network links from a colo to cloud provider
- Number of colos
- Number of fabric cache/gateway boxes
- Amount and performance of regional internal cache

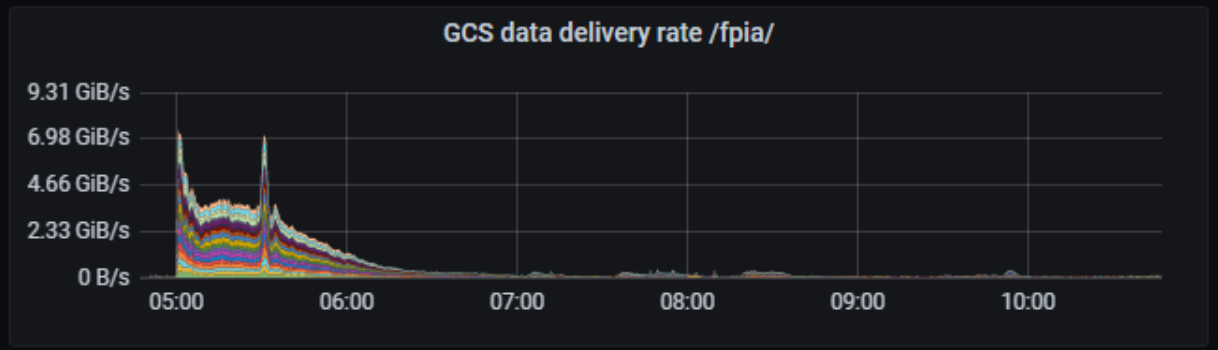


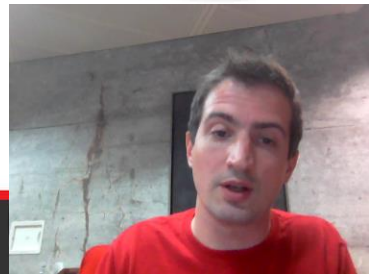


L3
(Varnish)

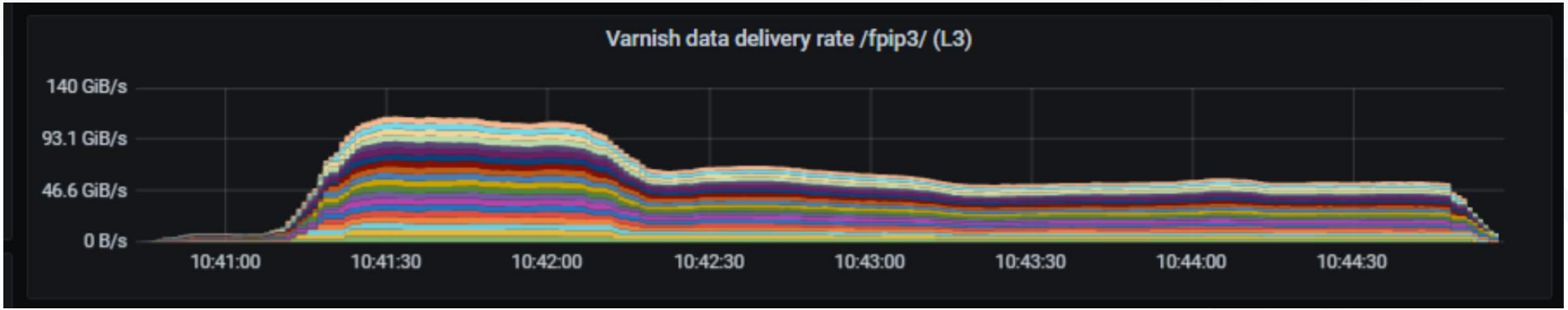


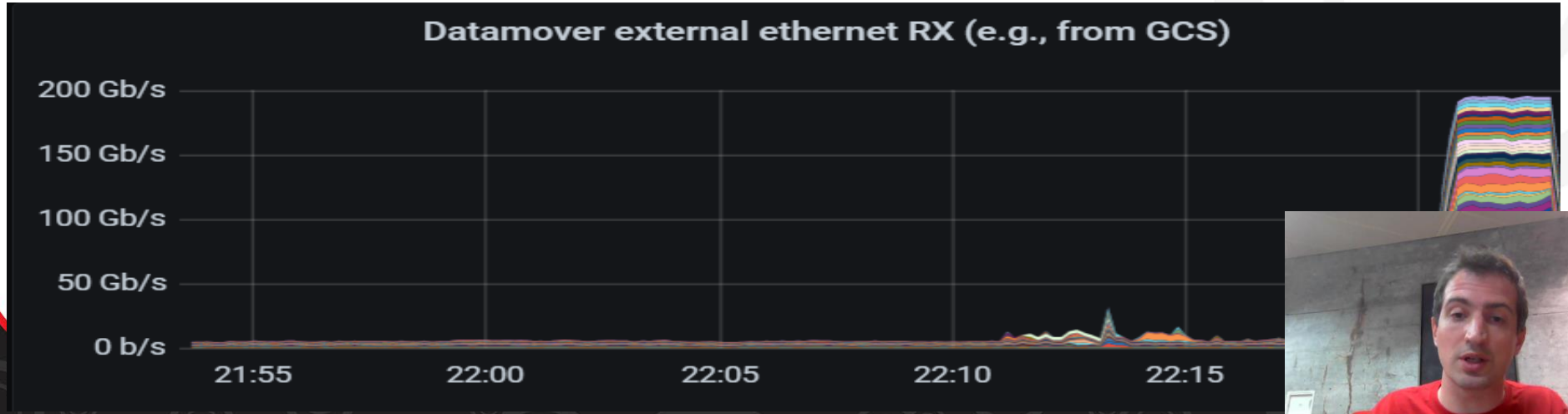
L4
(Vast)



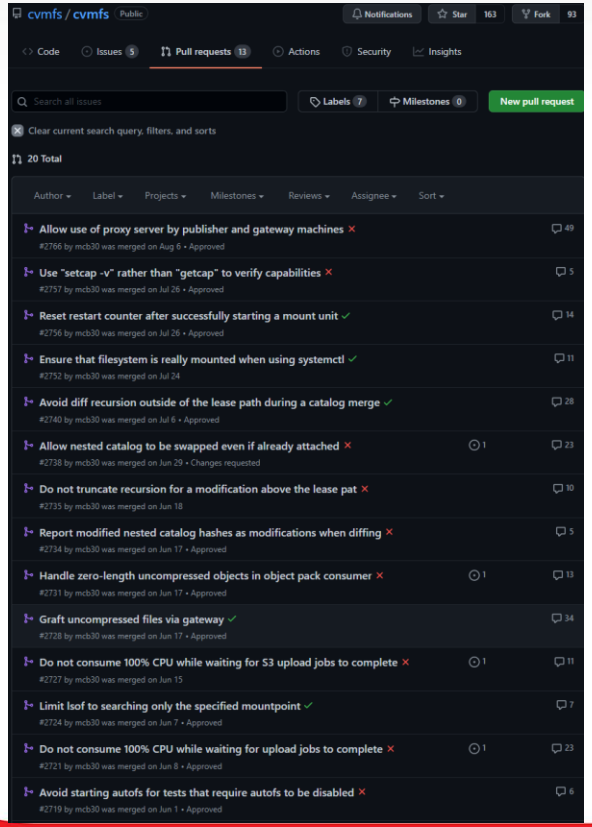


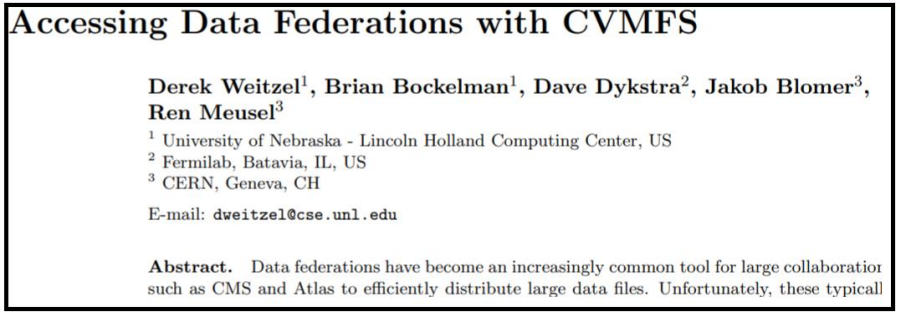
Stress testing





Write path details





Accessing Data Federations with CVMFS

Derek Weitzel[1], Brian Bockelman[1], Dave Dykstra[2], Jakob Blomer[3], Ren Meusel[3]

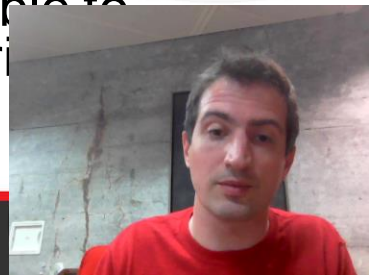
[1] University of Nebraska - Lincoln Holland Computing Center, US
[2] Fermilab, Batavia, IL, US
[3] CERN, Geneva, CH

E-mail: dweitzel@cse.unl.edu

Abstract. Data federations have become an increasingly common tool for large collaboration such as CMS and Atlas to efficiently distribute large data files. Unfortunately, these typicall

Weitzel et. al approach:

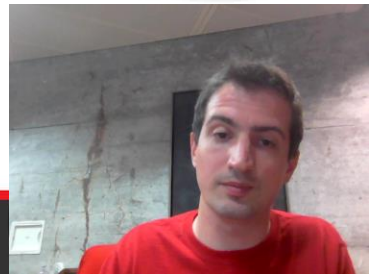
- CVMFS config tweaks to improve write scalability
- CVMFS “grafting” to separate data and metadata write paths
  - Data is uploaded directly to cloud storage
  - Metadata commits are batched together via CVMFS publishers and gateway
- “…more work is needed to be able to generate the graft files in a distri manner”. We took on this task.



Summary

- Moving to object store + caching allows us to scale in all directions for many years
- Using CVMFS to convert a POSIX read path into object store allowed us to keep our existing infra
- Well proven open source software like Varnish and commodity servers allows us to scale to hundreds of gigabytes per second IO

- Sound like fun? Jump is hiring.



Questions?