Exascale data processing with CVMFS

12 September 2022

Matt Harvey

HPC Production Engineer

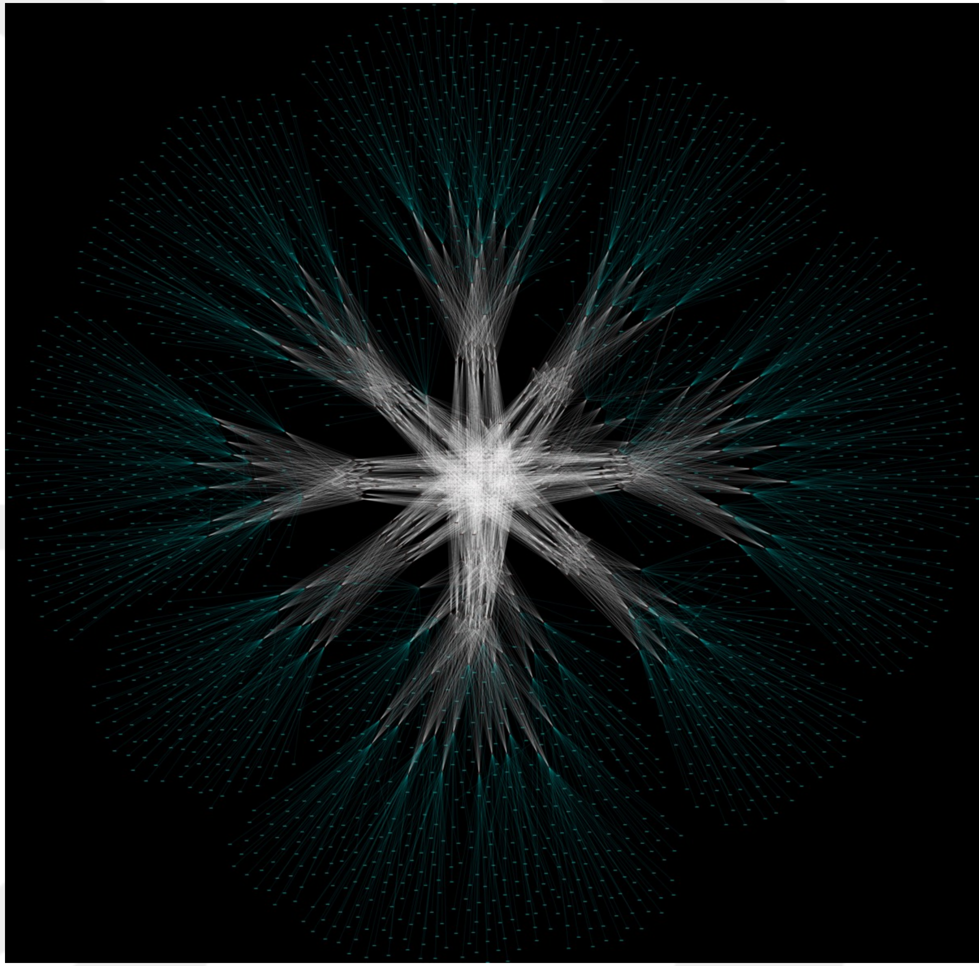
JUMP TRADING

- Privately-owned proprietary trading firm, established 1999
- Focus on algorithmic and high-frequency trading
  - Futures, options, equities, etc
- Significant investments in crypto infrastructure
- World-wide operations
  - 14 offices across US, EU, Asia, Pacific
- ~700 employees

HPC at Jump

Jump’s Research Environment (HPC / “The Grid”)

- The platform where we develop and optimize trading strategies
- Technologically competitive with some of the largest publicly known research systems in the world
- Thousands of servers
- Hundreds of petabytes of storage
- Fast network interconnects
- Sophisticated data-intensive and compute-intensive research workflows
- Keeps growing: more hardware every year



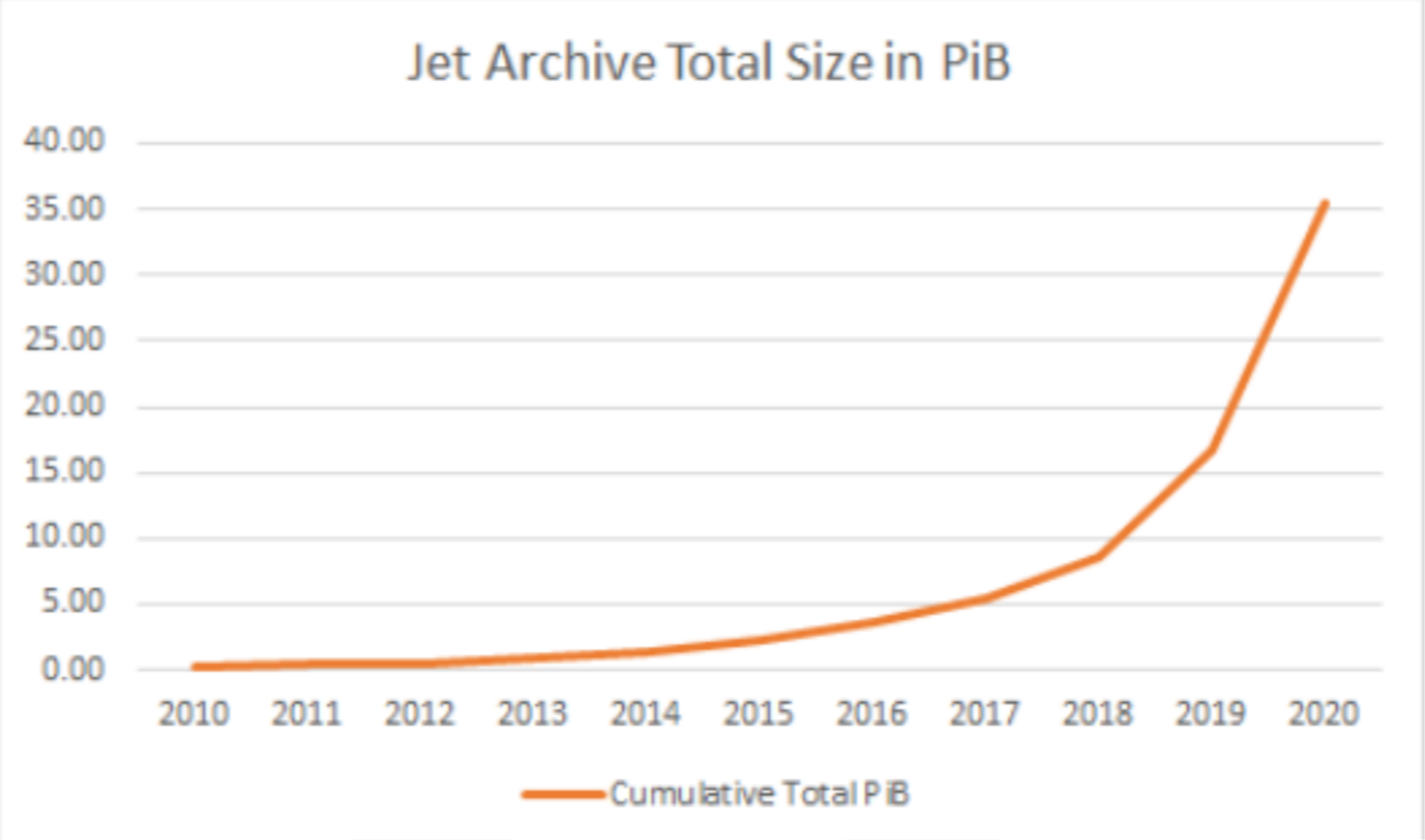
Fabric logical diagram
Image Credit: Olli-Pekka Lehto

DATA INTENSIVE

- Capture large volumes of market data, globally
- Daily processing to produce data products for quants
- Trading teams use it to regularly refit models
- Researchers use it to develop new strategies
- Typically large map-reduce workflows

Archive Growth History

Ten Years of Archive Growth



Archive Data Added Per Week

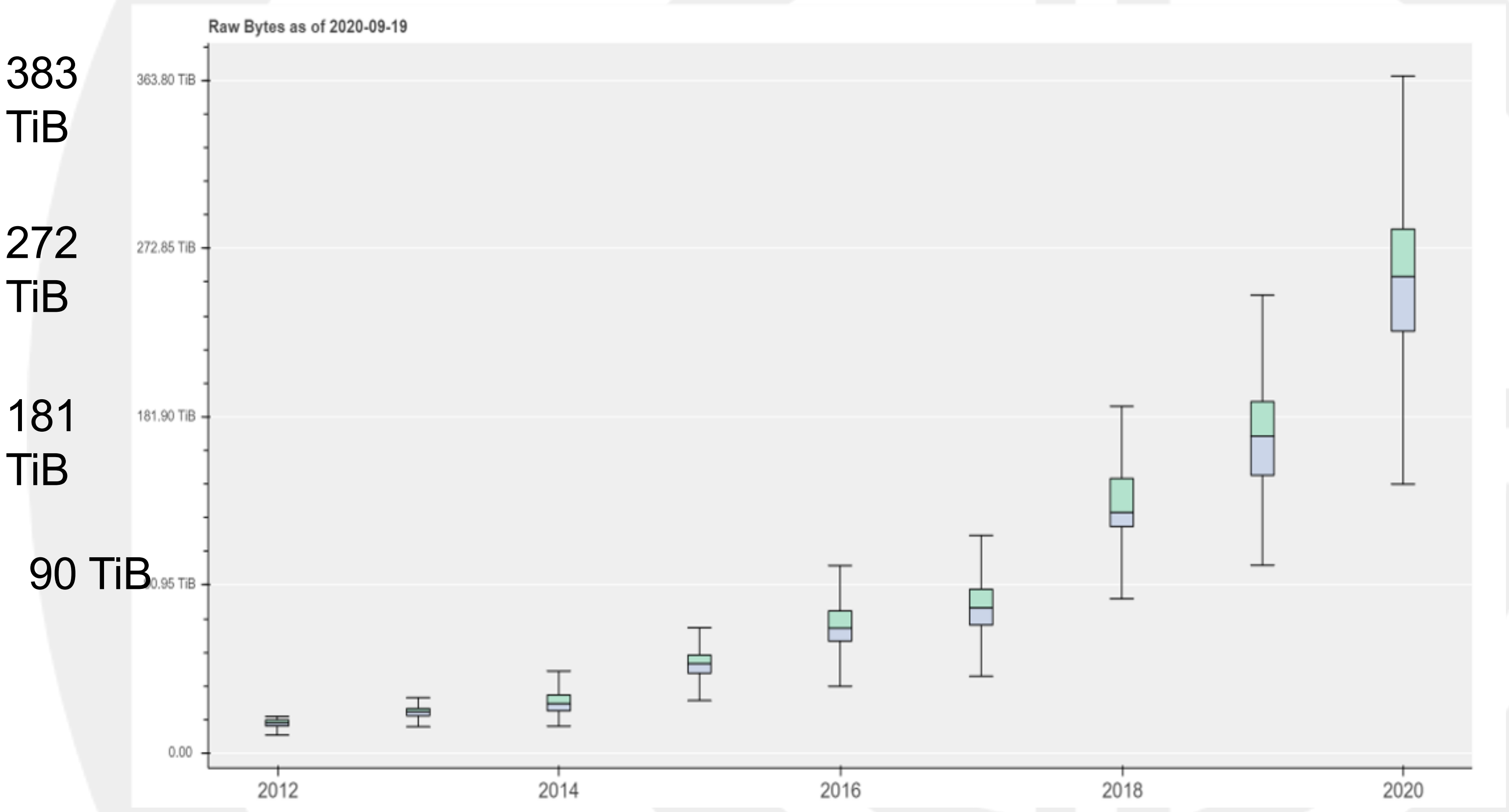
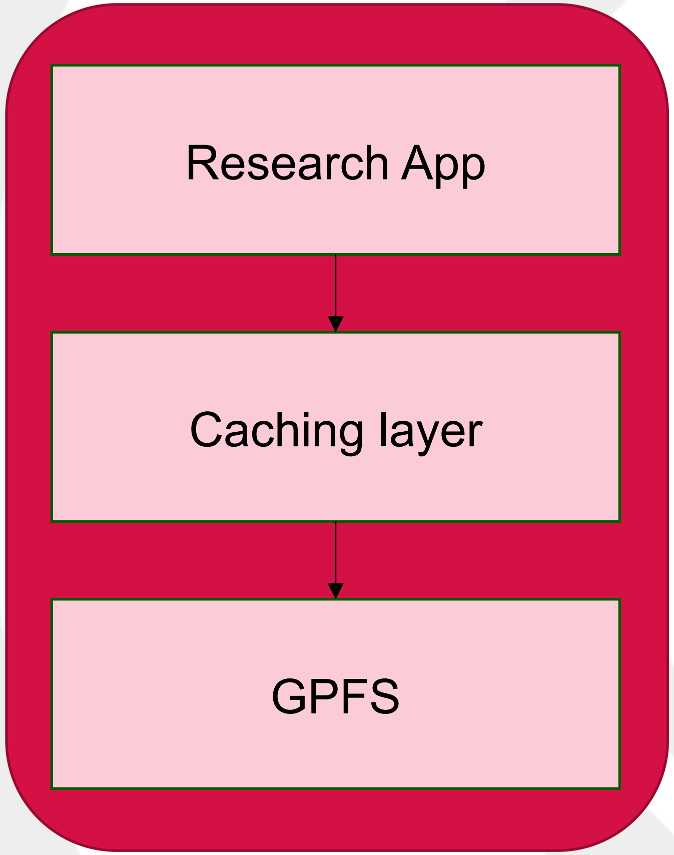


Chart and data courtesy of Tom Karas

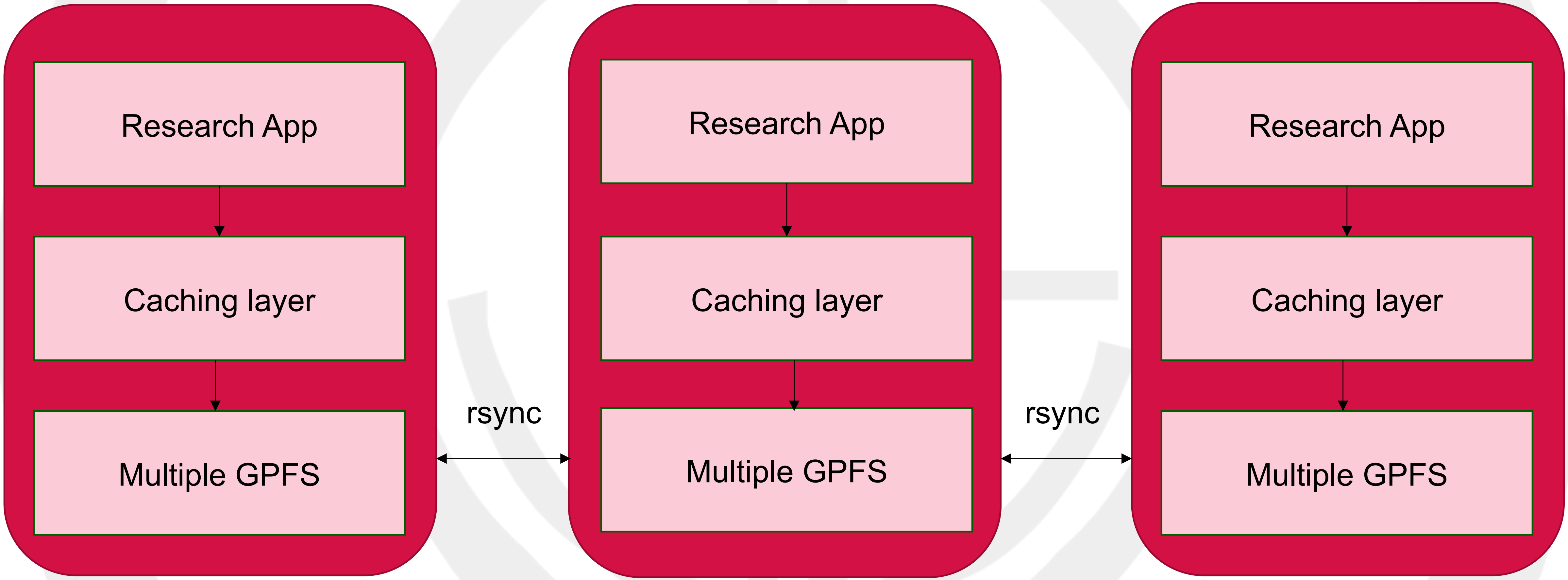
Description	Size
Total Bytes captured in 2020	9.75 PiB
Minimum Bytes captured per week in 2020	138.7 TiB
Maximum Bytes captured per week in 2020	441.9 TiB
Annual growth rate	1.5-2x prior year

Archive Infrastructure 2015



- A single large HPC system
- Lots of spinning disks
- GPFS filesystem

Then it grew..



..and lots of cold data living on disk..

The Legacy Archive

The Legacy Archive was an obstacle to growth.

- Paying for hard drives for data we never access (e.g., second copies)
- Very difficult to handle HPC pods scaling – choices are multihoming or more rsyncs. Neither are scalable.
- Massive rsyncs hard to manage
- Complex symlink and mount management to present 4+ GPFS/NFS filesystems as "/jump/archive"
- Hard to provide archive access in "non-grid" areas (e.g., dev hosts and hosts with experimental hardware)
- Hard to add new datasets: more hard drives, servers, and rsyncs
- GPFS is not the ideal place to store zero-bitrot data for decades

A New Design: Tiered Archive

- Able to run existing work-loads unmodified
  - POSIX FS presentation
- Decoupled from HPC fabric / filesystems
  - Accessible outside of HPC environment
- Able to accommodate >10x growth in size
  - Horizontally and vertically
- Non-requirements:
  - Read-write mounts on compute nodes
  - Concurrent writes on the same file
  - Fabric-wide consistency and file locking (unnecessary complexity and slowness)

The Tiered Archive

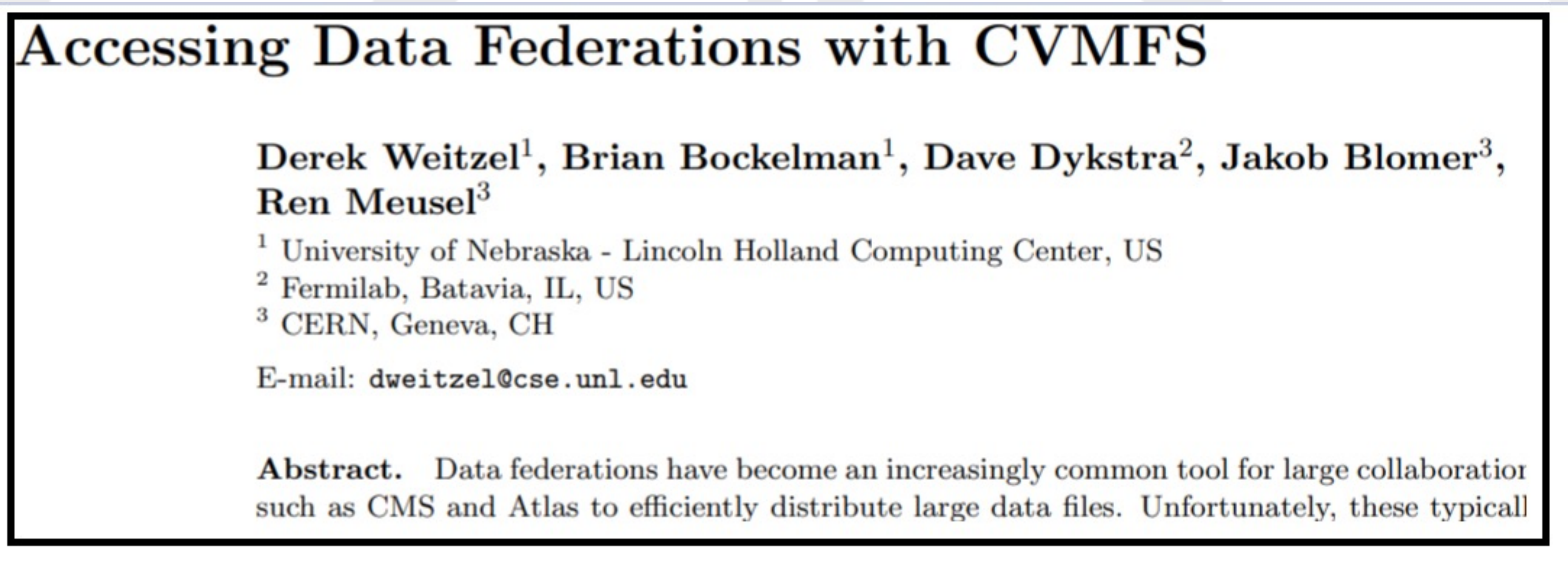
Three puzzle pieces:

1. **Read Path:** Cloud storage backed by many layers of cache

2. **Filesystem presentation:** POSIX-like to allow existing apps to keep working

3. **Write Path:** Pipeline to add new data to the system at scale

Tiered Archive: CVMFS



Accessing Data Federations with CVMFS

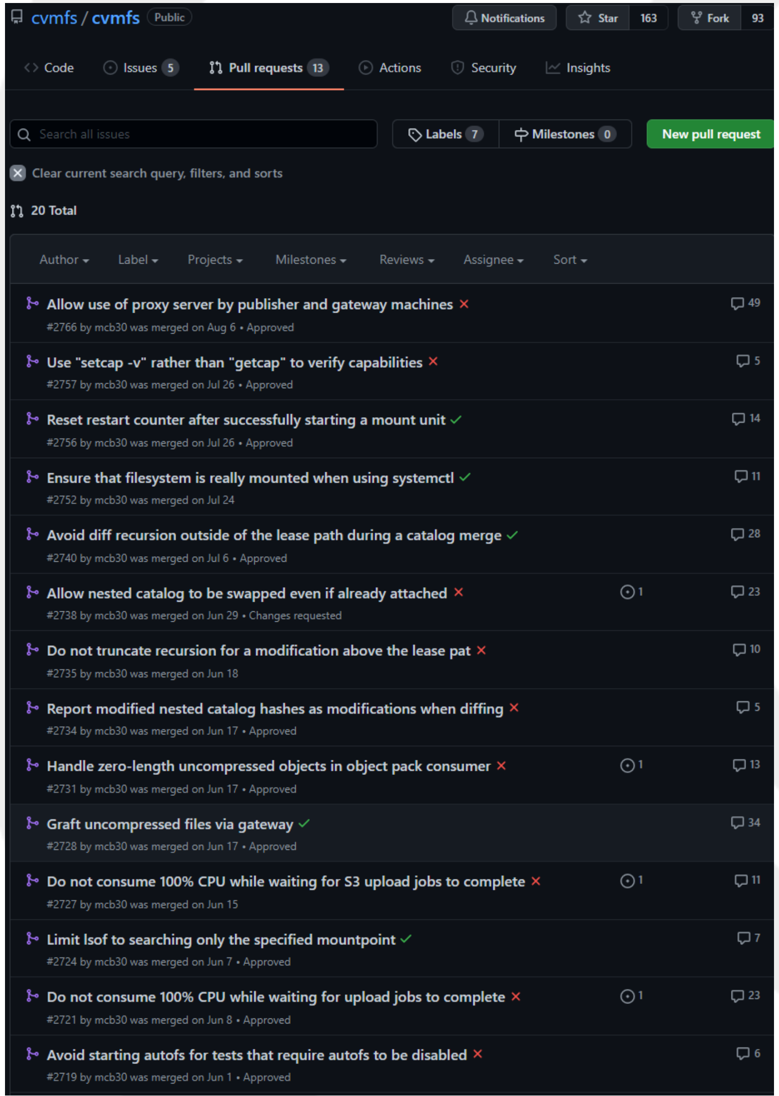
Derek Weitzel[1], **Brian Bockelman**[1], **Dave Dykstra**[2], **Jakob Blomer**[3], **Ren Meusel**[3]

[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 collaboratior such as CMS and Atlas to efficiently distribute large data files. Unfortunately, these typicall

Write path details



20 Total

- Allow use of proxy server by publisher and gateway machines ✗ — 49
  #2766 by mcb30 was merged on Aug 6 • Approved
- Use "setcap -v" rather than "getcap" to verify capabilities ✗ — 5
  #2757 by mcb30 was merged on Jul 26 • Approved
- Reset restart counter after successfully starting a mount unit ✓ — 14
  #2756 by mcb30 was merged on Jul 26 • Approved
- Ensure that filesystem is really mounted when using systemctl ✓ — 11
  #2752 by mcb30 was merged on Jul 24
- Avoid diff recursion outside of the lease path during a catalog merge ✓ — 28
  #2740 by mcb30 was merged on Jul 6 • Approved
- Allow nested catalog to be swapped even if already attached ✗ — 1 — 23
  #2738 by mcb30 was merged on Jun 29 • Changes requested
- Do not truncate recursion for a modification above the lease pat ✗ — 10
  #2735 by mcb30 was merged on Jun 18
- Report modified nested catalog hashes as modifications when diffing ✗ — 5
  #2734 by mcb30 was merged on Jun 17 • Approved
- Handle zero-length uncompressed objects in object pack consumer ✗ — 1 — 13
  #2731 by mcb30 was merged on Jun 17 • Approved
- Graft uncompressed files via gateway ✓ — 34
  #2728 by mcb30 was merged on Jun 17 • Approved
- Do not consume 100% CPU while waiting for S3 upload jobs to complete ✗ — 1 — 11
  #2727 by mcb30 was merged on Jun 15
- Limit lsof to searching only the specified mountpoint ✓ — 7
  #2724 by mcb30 was merged on Jun 7 • Approved
- Do not consume 100% CPU while waiting for upload jobs to complete ✗ — 1 — 23
  #2721 by mcb30 was merged on Jun 8 • Approved
- Avoid starting autofs for tests that require autofs to be disabled ✗ — 6
  #2719 by mcb30 was merged on Jun 1 • Approved

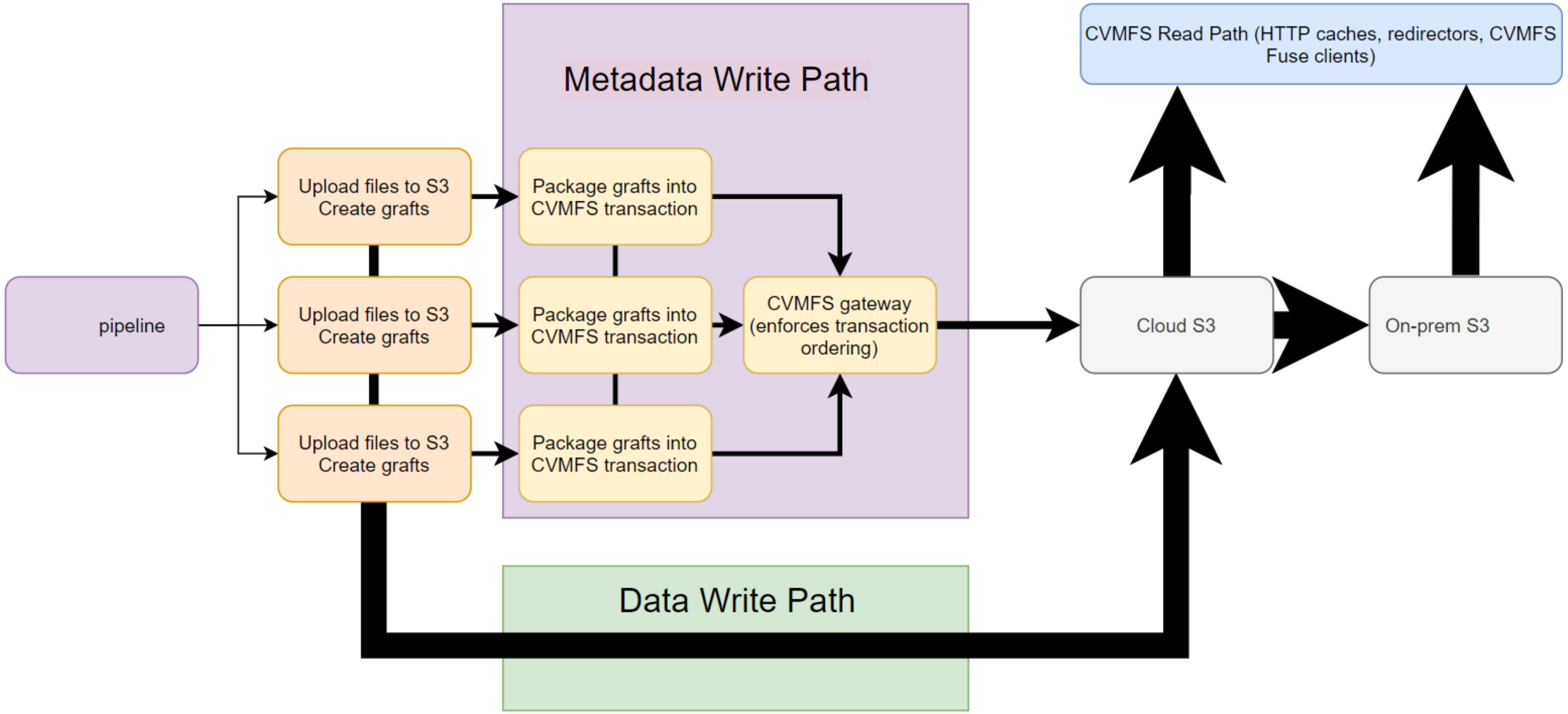
Weitzel et. al approach:

- 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 distributed manner”. We took on this task.

Archive Write Path

- Workflow pipeline manages publication

- Uploader
  - Upload objects
  - Create graft files
- Grafter
  - Process graft batches
  - Automate CVMFS transactions
  - Fleet of CVMFS publisher containers
- Staging Area (NFS) exported from VAST appliance
  - Scratch space for data generation
  - “just in time” data access


CVMFS Read Path (HTTP caches, redirectors, CVMFS Fuse clients)
Metadata Write Path
pipeline
Upload files to S3 Create grafts
Upload files to S3 Create grafts
Upload files to S3 Create grafts
Package grafts into CVMFS transaction
Package grafts into CVMFS transaction
Package grafts into CVMFS transaction
CVMFS gateway (enforces transaction ordering)
Cloud S3
On-prem S3
Data Write Path

Archive Read Path

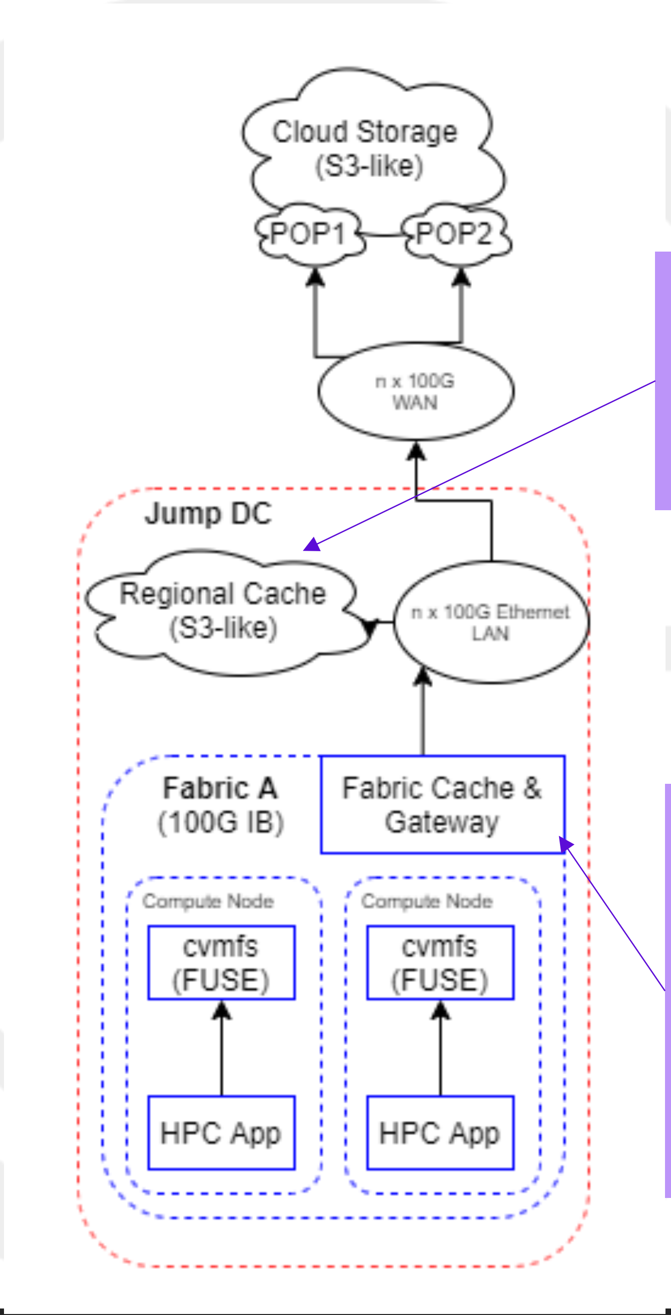
2xPOPs each with 2x100G

100G leaf/spine
Ethernet network

HDR (1 or 200G IB) fabric

CVMFS FUSE mount

Unmodified application



Google Cloud Storage

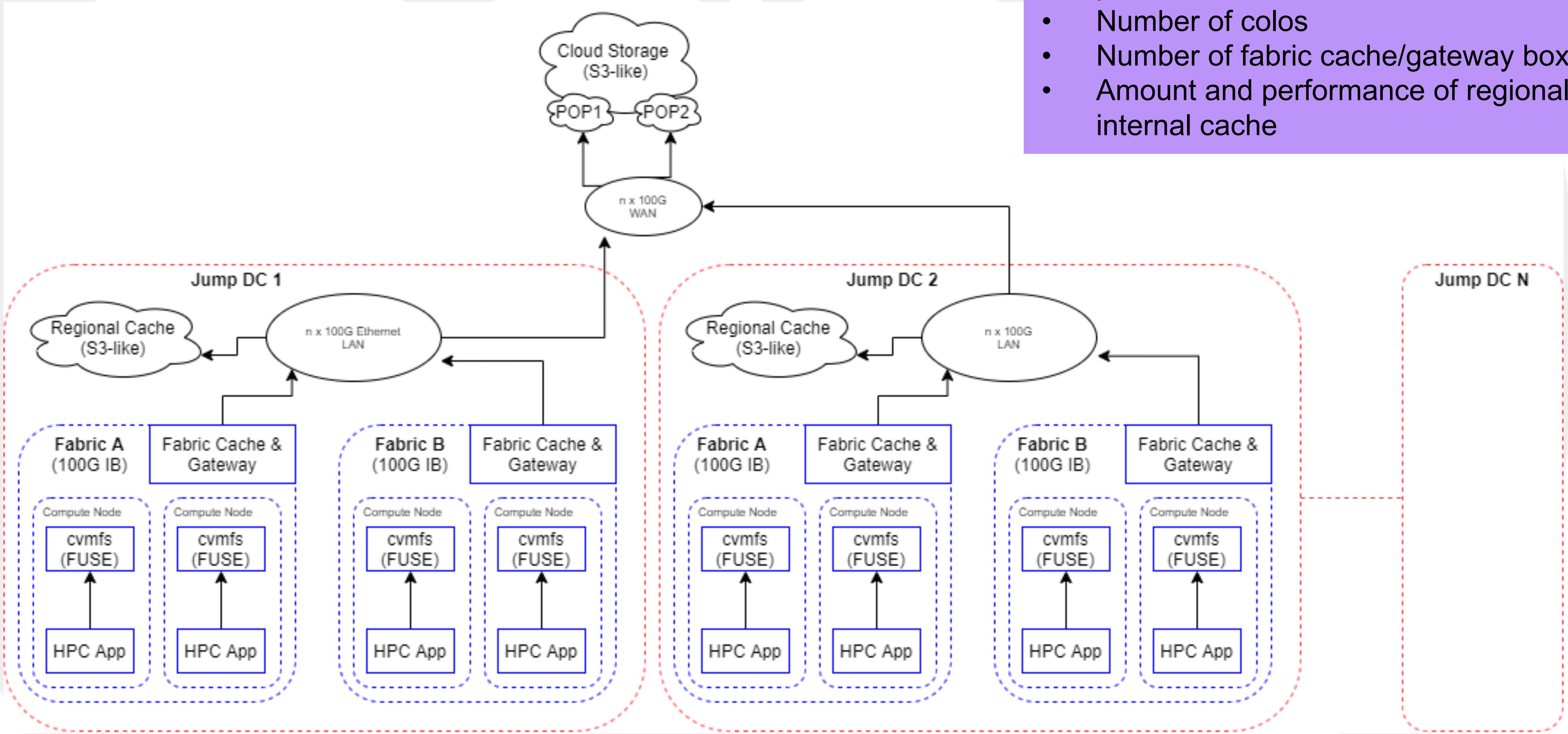
~5PB VAST S3 Appliance
Mix of 3D XPoint (fast) and
QLC (slow/economical)

Cache/Gateway box
- Many NICs IB/Eth
- ~40T SSD
- Standard Linux
- Varnish Plus

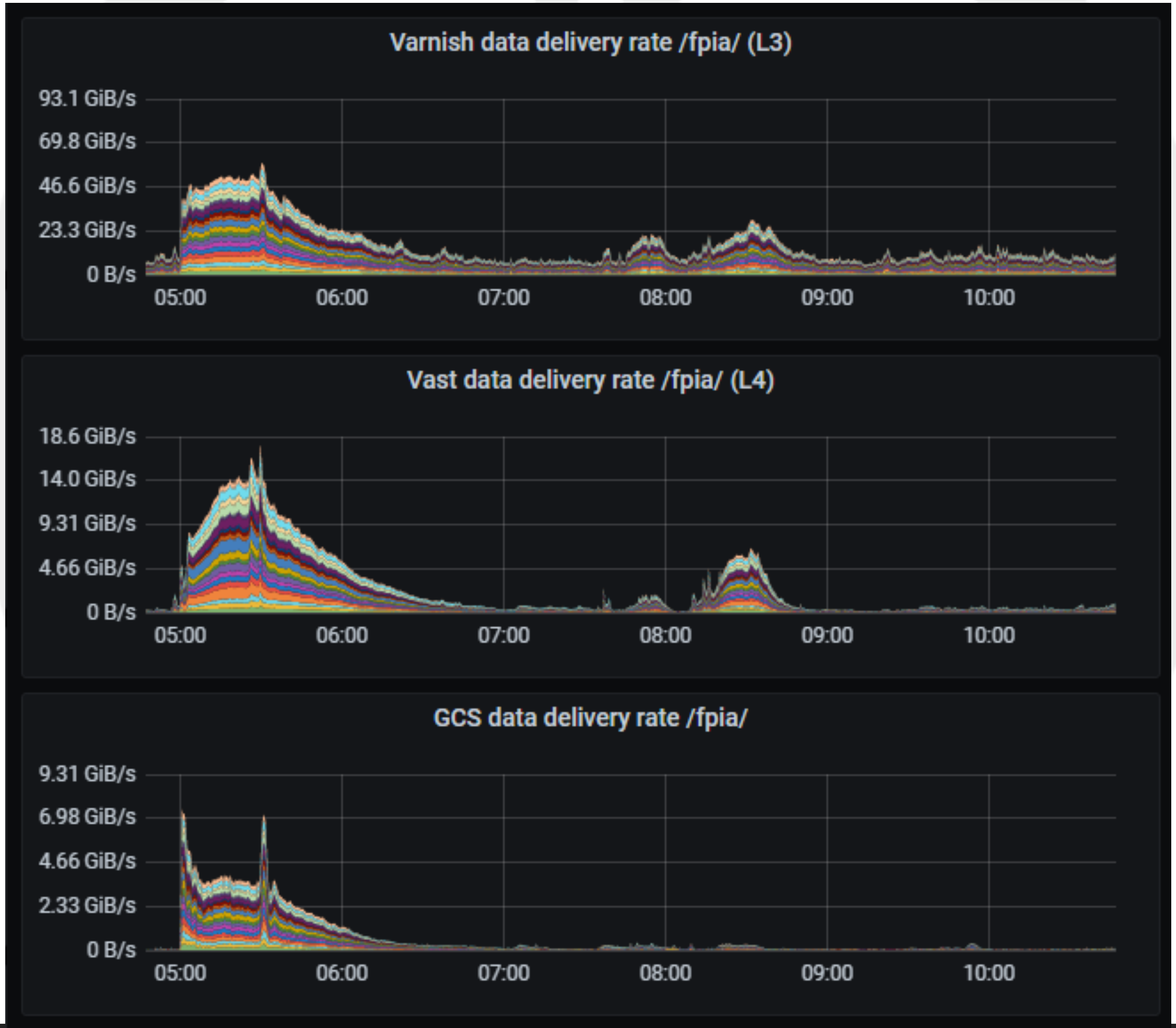
Designed for the next 10 years

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

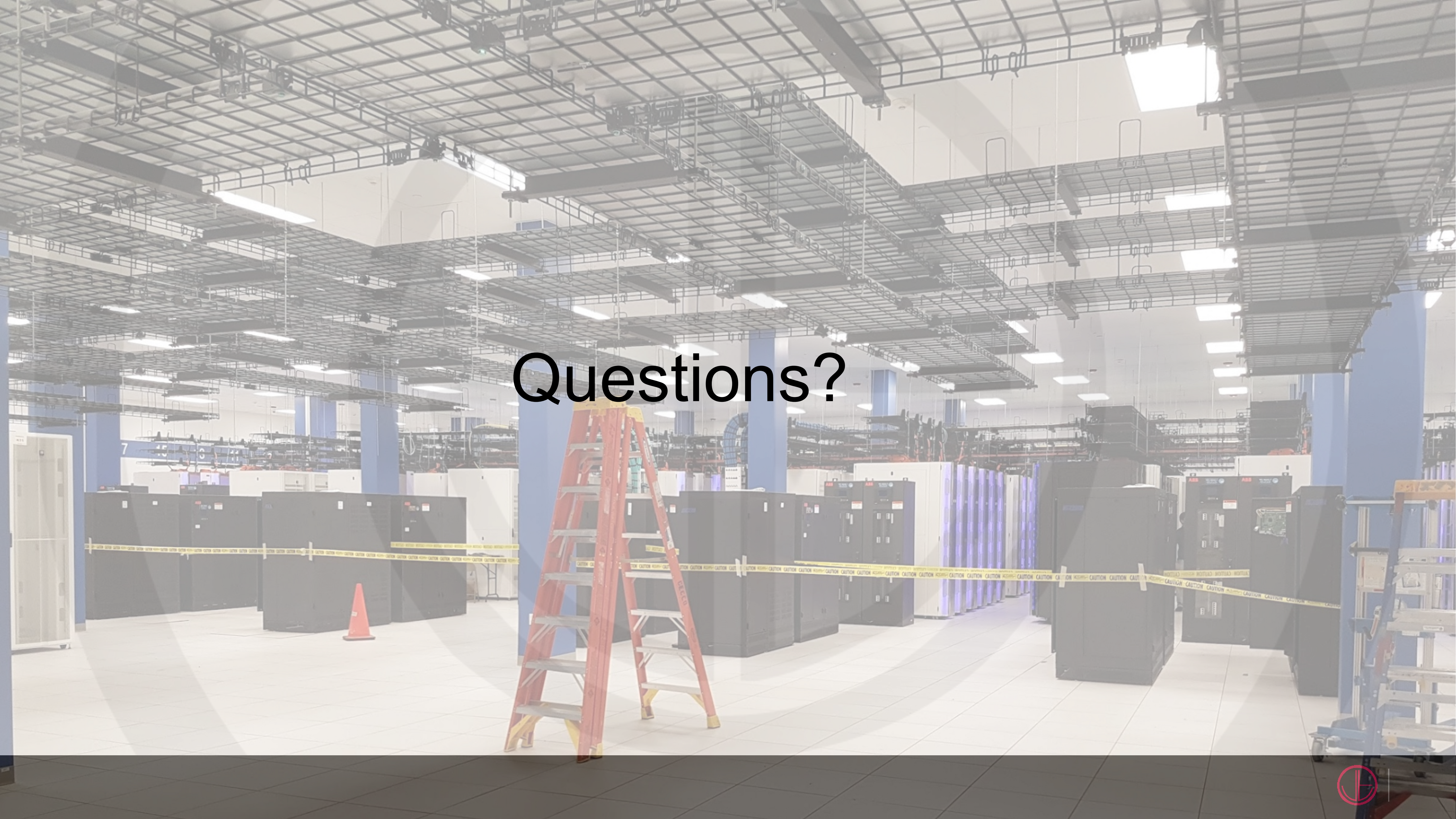


Performance



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?