IRIS-HEP 200Gbps challenge

WLCG/HSF
Workshop

Brian Bockelman

MORGRIDGE
INSTITUTE FOR RESEARCH

Credit Where it is Due

- This presentation summarizes a large body of work across IRIS-HEP, USATLAS, and USCMS.

- Fermilab: Lindsey Gray, Nick Smith
- Morgridge: Brian Bockelman
- Notre Dame: Ben Tovar
- Princeton: Jim Pivarski
- U. Chicago: Lincoln Bryant , Rob Gardner, Fengping Hu, David Jordan, Judith Stephen , Ilija Vukotic
- National Center for Supercomputing Applications: Ben Galewsky
- U. Nebraska: Sam Albin, Garhan Attebury, Carl Lundstedt, Ken Bloom, Oksana Shadura, John Thiltges, Derek Weitzel, Andrew Wightman
- UT-Austin: KyungEon Choi, Peter Onyisi
- U. Washington: Gordon Watts,
- U. Wisconsin: Alex Held, Matthew Feickert

WLCG Data Challenges

- **The recently-completed DC24 (and the DC21 predecessor) showed community readiness at 25% of HL-LHC scale**.
  - That's a powerful statement!
- Why else is this a remarkable success? These challenges are:
  - **Are integrative**: Brings together software providers, services, and facilities. A vertical stack that's difficult to coordinate across the business of "everyday life".
  - **Deadline-driven**: Forces teams to to deliver and a clear evaluation point.
  - **Quantitative**: Enables measurement of progress, year-over-year.
- In a world full of details, the data challenges are help us communicate!

Grand Challenge as a Framework for Progress

- Within IRIS-HEP, we've used the concept of "Grand Challenges" to help drive progress in the project toward the HL-LHC.
  - We define these to be a series of incremental, increasingly-realistic exercises toward a common goal.
- What makes them so useful?
  - Focuses effort
  - Helps the community find "common truths".
  - Can include both scale and technology readiness.

If it works at 10X, then we understand it better at 1X!

DC: Scale *and* Technology Readiness

- Around the same time as DC21, we'd been working within WLCG DOMA to introduce HTTP-TPC as a transport technology.
  - We felt it was ready.
- **Problem: How do we show the community HTTP is ready?**
  - **Solution: DC21!** Use the data challenges as a staging ground for showing new ideas.

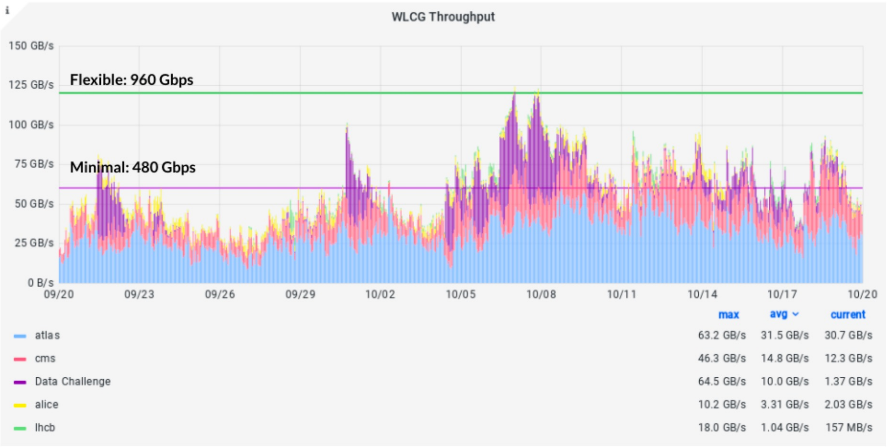


Figure 1 - Mock DC1 22/09/2021; Mock DC2 01/10/2021; Network Challenge (DC) 04-10/10/2021; Tape Challenge 11-19/10/2021.

Transfer scaling during DC21.
Figure reproduced from
https://zenodo.org/record/5767913

DC: Scale *and* Technology Readiness

- **Happy ending!**
- DC21 showed that HTTP was viable for replacing GridFTP at LHC scales.
- Community adoption & uptake was rapid.
  - By the end of 2021, nearly all bulk data transfers for LHC migrated to the new protocol.

- Not all technologies will have happy endings.
- Important piece is using 'grand challenges' to move the community forward.

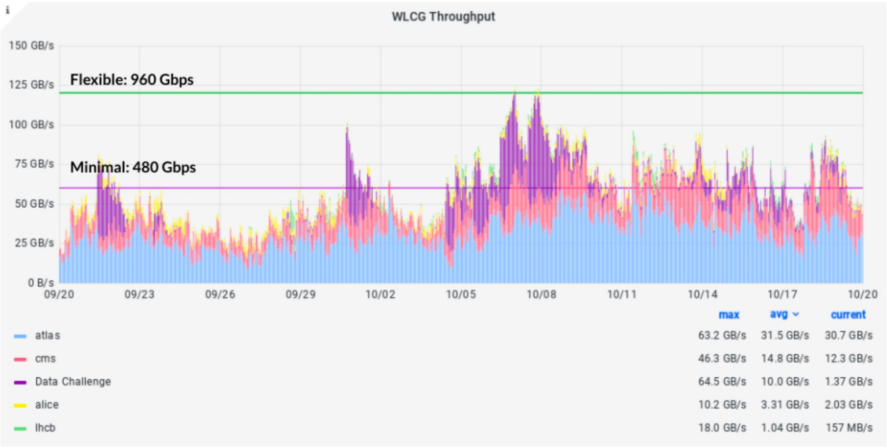


Figure 1 - Mock DC1 22/09/2021; Mock DC2 01/10/2021; Network Challenge (DC) 04-10/10/2021; Tape Challenge 11-19/10/2021.

Transfer scaling during DC21.
Figure reproduced from
https://zenodo.org/record/5767913

Grand Challenge as a Framework for Progress

- The "Grand Challenge" approach has been instrumental in focusing the community and the institute.
  - It's applicable to both scale and technology readiness.

Idea: Let's do the same thing for "analysis at HL-LHC scale"

The 200Gbps Challenge

- **Observation**: IRIS-HEP innovates in
  - Facilities R&D (how do we build better compute facilities for HL-LHC; SSL area).
    - Includes pathfinder facilities that can access ATLAS, CMS, or open data.
    - These facilities partner with existing, large T2 sites (T2_US_Nebraska, MWT2); done purposely so one could scale for tests.
  - Analysis systems (bringing the Python-based analysis ecosystem in production).
  - Data delivery (effective delivery of events to compute).

- **Idea (13 March @ Chicago)**: Pull the three efforts together and show readiness at 25% of HL-LHC scale.
  - And, 20 March @ CERN, we came up with the idea of presenting results (here) at the WLCG Workshop in May 2024. **7 weeks to execute!**

25% of what, exactly?

- We want to show significant, quantitative progress toward HL-LHC-scale analysis.
  - Like in DC21, use realistic proxies for HL-LHC.
- In DOMA, we were able to tap into a long history of facility planning and was able to get the community to agree to goals based on extrapolating from a decades-old system.
  - No such luck in analysis. **Very little agreement** on HL-LHC analysis models.

- We decided to put down our own axioms for the challenge:
  1. We believe a full-scale HL-LHC analysis requires high-data rates, **reading 200TB in 30 minutes**.
  2. We want to use the IRIS-HEP Data Analysis pipeline and SSL facilities.

- Longer-term, we're trying to socialize the need for the community to find common truths.

Why 200TB in 30 minutes?

- Why select X TB in Y minutes?   (X=200, Y=30)

For an independent calculation that arrived at a similar conclusion, see [L. Gray's ACAT 2024 talk].

- Experience shows we hit scaling limitations when we go up by an order of magnitude.
  - Running smoothly at 10X brings immediate benefit back to the 1X case.
  - If we fail to run smoothly at 10X then we gain valuable insight into the current limitations.
- This is ambitious-but-realistic for extrapolating today's facilities out 4 years.
  - There's nothing exotic or out of the reach of a typical US T2 in the 2028 timeframe.
- This is within reason by extrapolating today's parameters out to the HL-LHC event counts and sizes.
  - There's no first-principles derivation of the leading order. One also cannot argue that missing these targets will cause HL-LHC to fail.
    - But then again, the same is true for DC24.

Points to the need for 'common truths' in the community around HL-LHC analysis

Derived Values – Example CMS ‘napkin math’

- Start with 200TB read in 30 minutes. => ~900Gbps sustained.
- 25% scale => 200Gbps sustained. Hence, **200Gbps challenge**.

- 200Gbps over 30 minutes => 45TB of data into the analysis process.
- Assume 25% of the data read from the CMS NanoAOD
  - => 180TB of NanoAOD is required to push 45TB of branches.
- At 2KB/event, 180TB of NanoAOD is **96B events**.
- 96B events in 30 minutes => sustained 55MHz event rate.

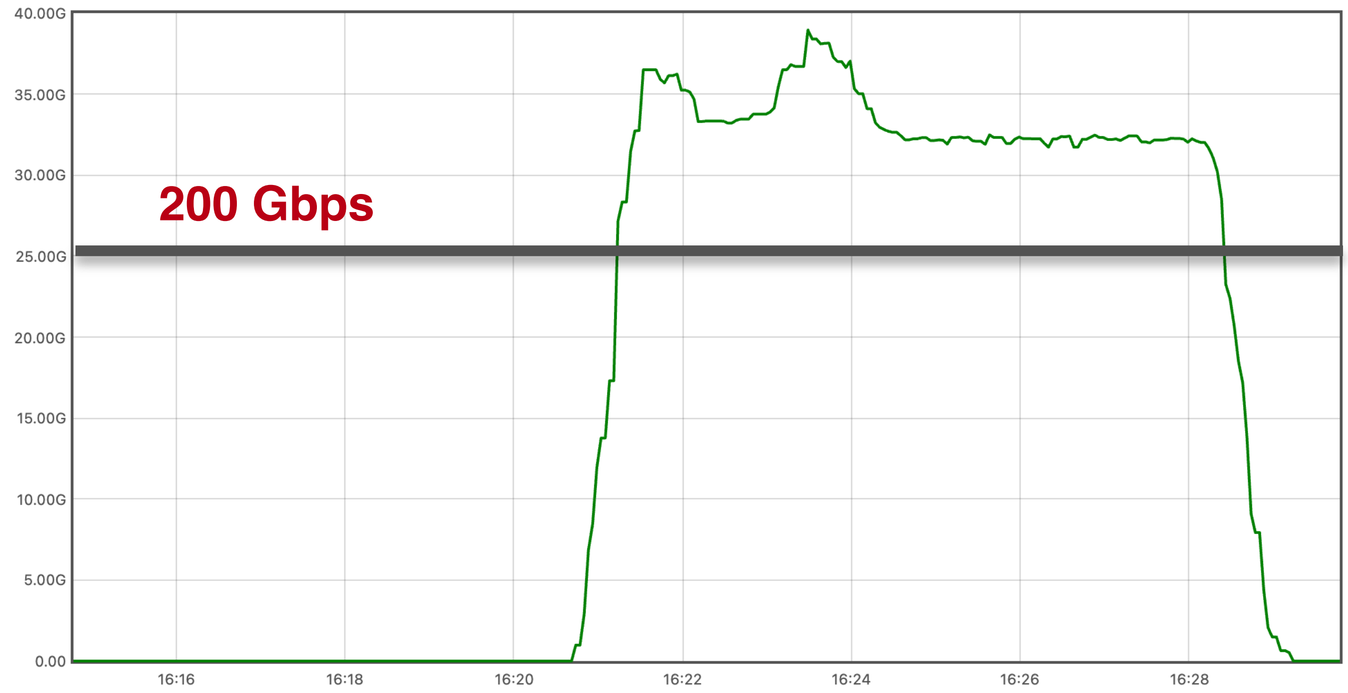
Our sample analysis runs at 25KHz per core, meaning 2,200 cores are needed to sustain the 55MHz event rate.

200Gbps Challenge – Strategies to completion

- Given we want realism (use real data, not Open Data), we split into two teams – one working with ATLAS PHYSLITE at Chicago, the other CMS NanoAOD at Nebraska.
  - The “napkin math” from prior slide was repeated for ATLAS
- **Immense**, focused activities across the institute.
- First week was focused on planning.
- Both facilities had to work to reprovision hardware to go into “test mode”.
  - Special credit to Chicago team who also reworked their network topology to provide more bandwidth for the test.
- In each case, we also had to be mindful of existing analysis & production activities.
- Progress was made: the graph to the right shows the performance of a clustered XCache service at the end of week 4.

Busy Slack even!

Name ^	Date created	Total membership	Messages posted
# 200gbps-challenge	2024-03-19	20	2,740
# 200gpbs-challenge-atlas	2024-03-29	19	2,580



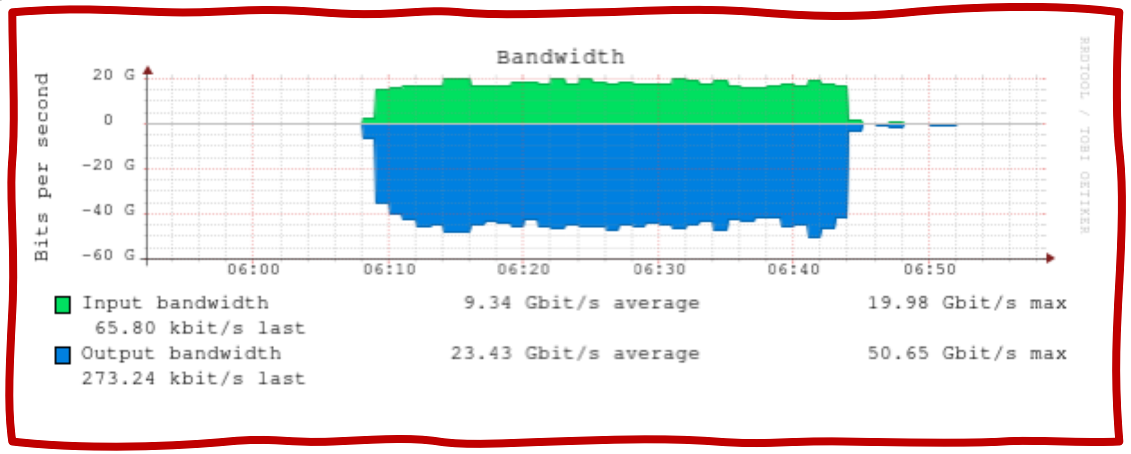
Facilities

FEARLESS SCIENCE

MORGRIDGE
INSTITUTE FOR RESEARCH

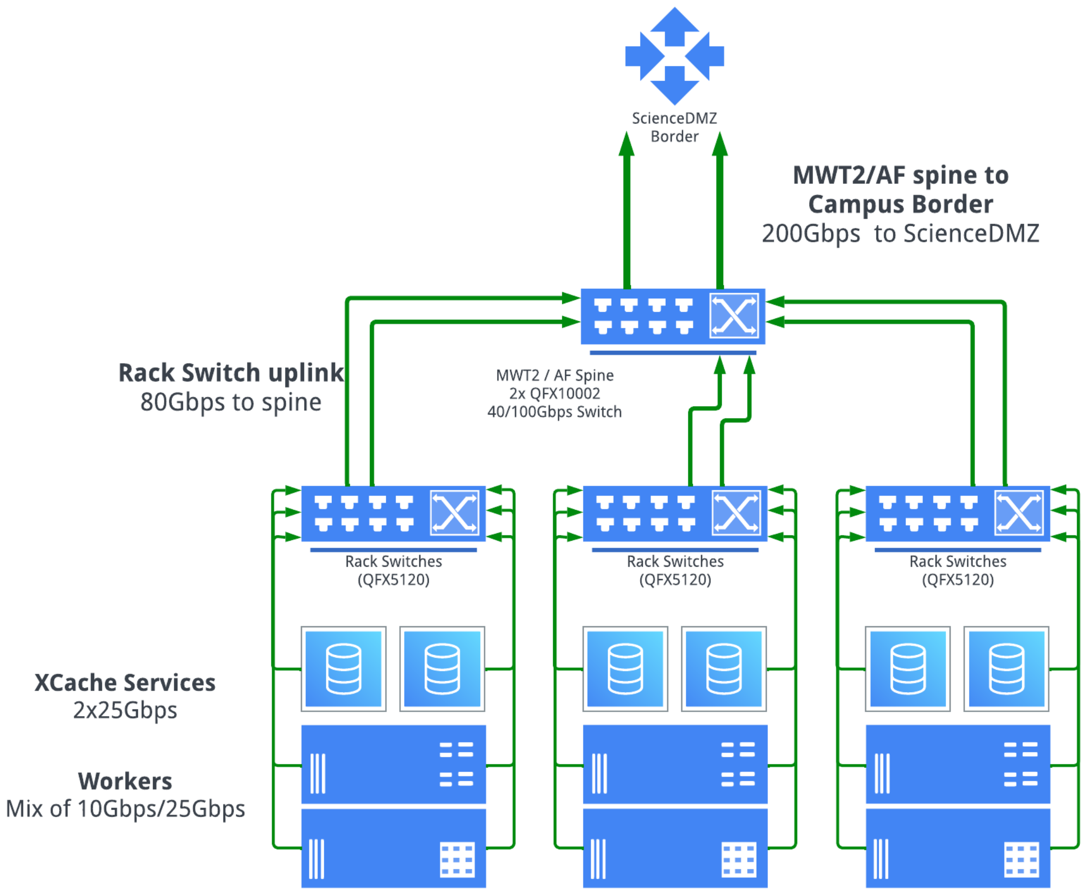
Common Ingredients – Shared Facilities, Kubernetes, XCache

- The 200 Gbps challenge activities leveraged both dedicated IRIS-HEP hardware and local T2 sites.
  - For the larger runs, temporarily repurposed worker nodes from the T2.
- Both Chicago and Nebraska use Kubernetes to launch and manage services.
  - Automates the network configuration.
  - Easy to rapidly iterate through service versions.
  - Useful for persistent services (e.g., JupyterHub, XCache) or transient workers.
- XCache was used as the storage technology.
  - This is the venerable XRootD daemon configured in a caching mode.
    - Data is pulled in on-demand from remote sites (Rucio for ATLAS or AAA for CMS).
    - Subsequent reads are from internal to the AF.
  - Both sites had 8 XCache hosts packed with NVMe.
    - Able to show ~45Gbps / host of throughput in dedicated testing with xrdcp/curl.



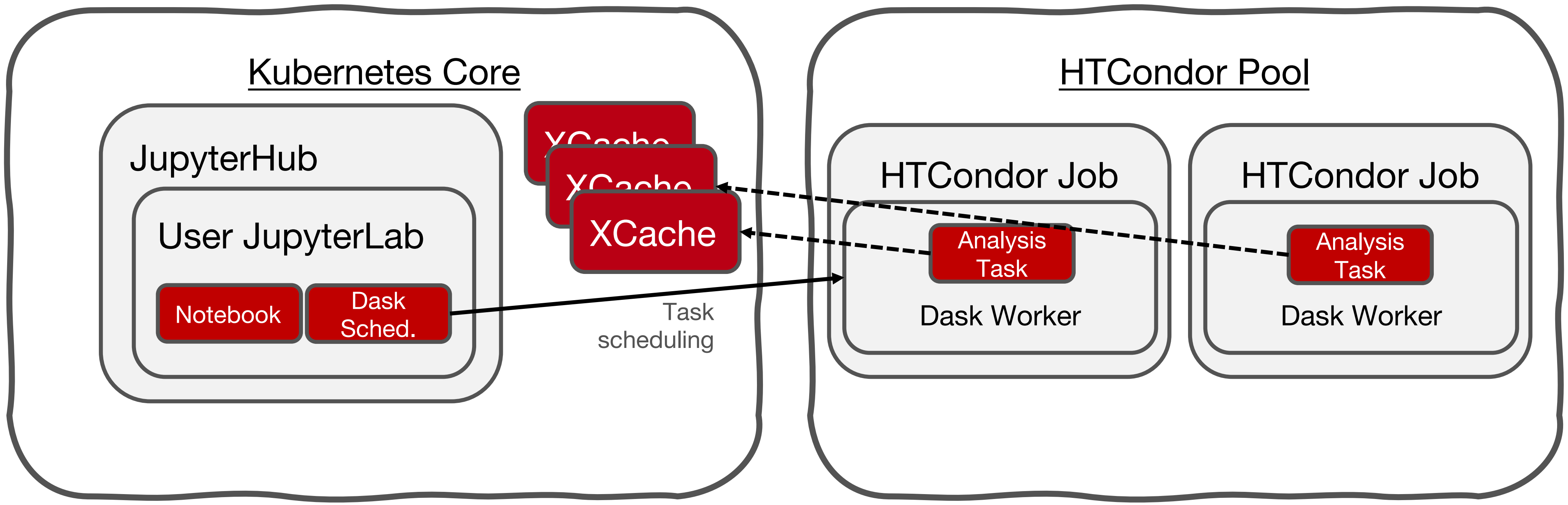
Chicago

- At Chicago, we partitioned the XCache hosts across multiple switches to maximize bandwidth.
  - Tricky network topology – some workers on same switch as XCache, some data went across network backbone.
- For the largest runs, used up to 2.5k cores.
  - All cores were used via Kubernetes
  - Tests were driven by scripts.



Nebraska

- Tests were driven via Jupyter notebook at the Coffea-Casa facility.
- Scale-out was done to the T2's HTCondor pool.
- All authorization done via tokens issued by CMS's IAM instance.
- Each of 2 Kubernetes switches uploaded to the network core via 2x100GbE.
  - TOR switches for HTCondor range from 2x40GbE to 6x40GbE to 2x100GbE.



Trying different Toolsets

FEARLESS SCIENCE

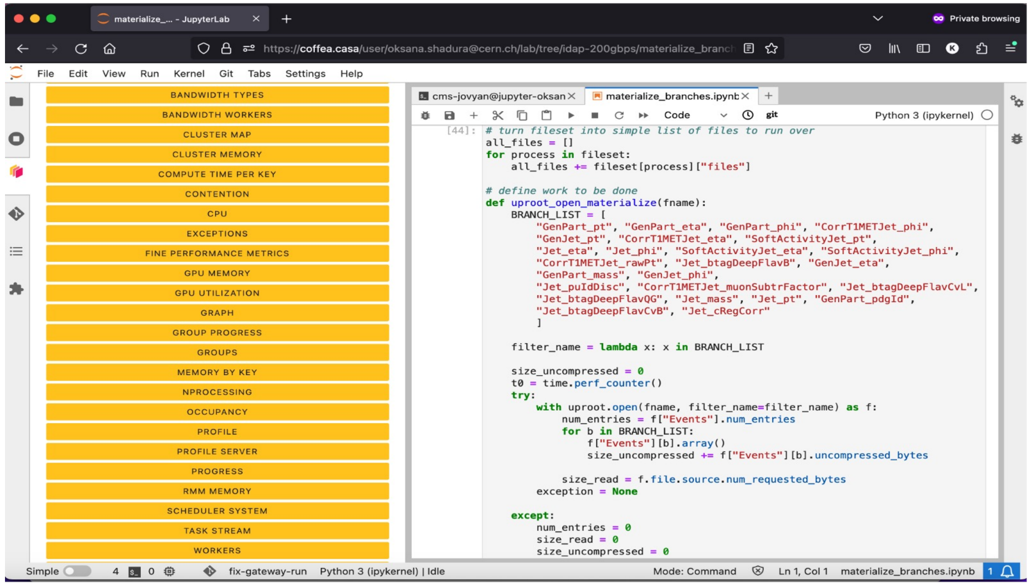
MORGRIDGE INSTITUTE FOR RESEARCH

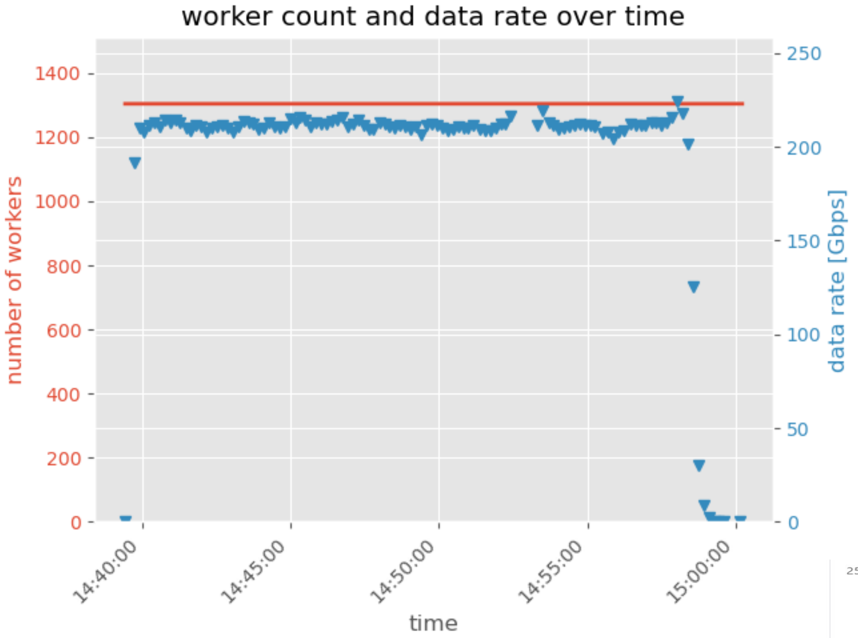
Uproot + Coffea Toolset

- For Uproot + Coffea, we decided:
  - Start with CMS Run2 NanoAOD (~100TB).
  - Process with Coffea 2024. Read data from XCache on the Coffea-Casa facility at the Nebraska Tier-2.
  - Start with the IDAP notebook from the AGC work last year, expand work out into the site HTCondor.
  - Dask tasks processed in TaskVine & Dask.
  - Compute values from the events read in; accumulate into histograms. “Direct from NanoAOD” style analysis.
- Notes on realism:
  - Real XCache setup. Token-based auth using the IAM service at CERN.
  - LZMA decompression dominates analysis time (~70%). To hit our target 25KHz-per-core processing rate, we recompressed the NANOAOD using ZSTD. About 20% larger than the original dataset, ~2.5x faster.
    - N.b.: our strong opinion is CMS needs to make this change.
  - We scale-out to HTCondor but, for these tests, pre-create the workers.
  - For at-scale tests, we dropped coffea and went straight uproot due to under-investigation memory issues.

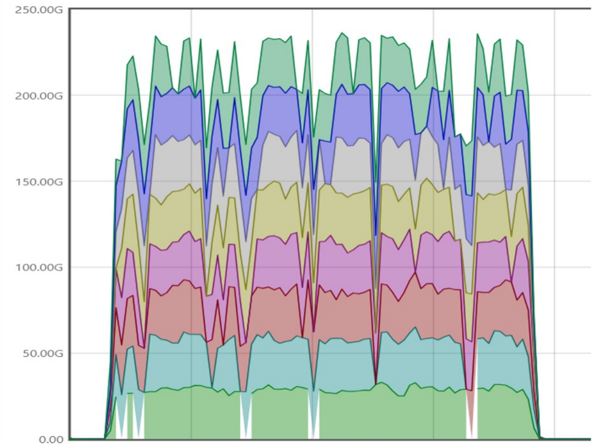
Uproot Results

- Highest data-rate configuration (TaskVine):
  - Data read (compressed): 58.33TB
  - Average data rate: 221Gbps
  - Peak data rate: 240Gbps
  - Files processed: 63,762 (17 failed)
- Highest event-rate configuration (Dask):
  - total event rate : 32,256 kHz
  - Processed 40,276,003,047 events total
  - Per-core event rate : 27.66 kHz



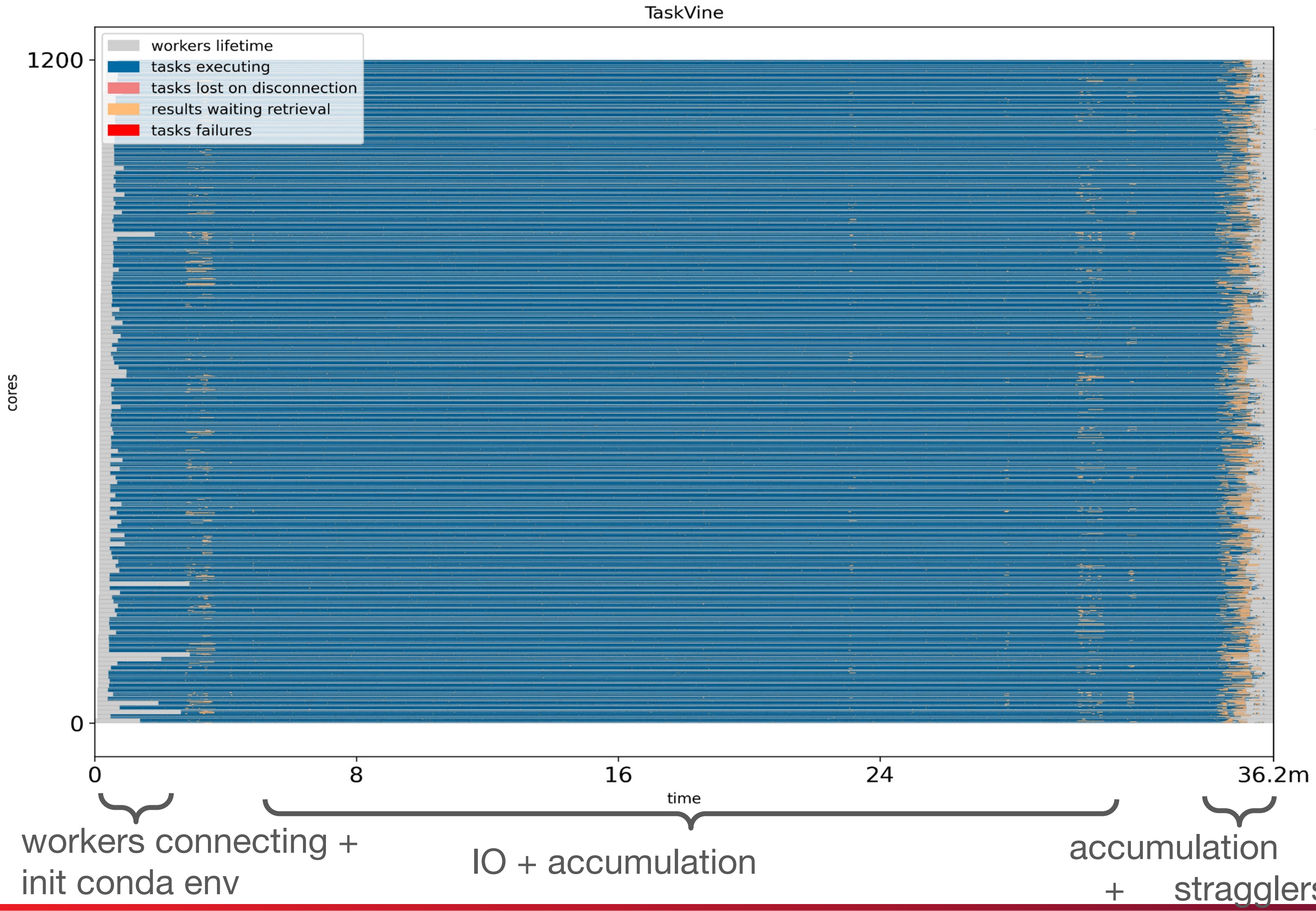


Network rates from XCache storage.

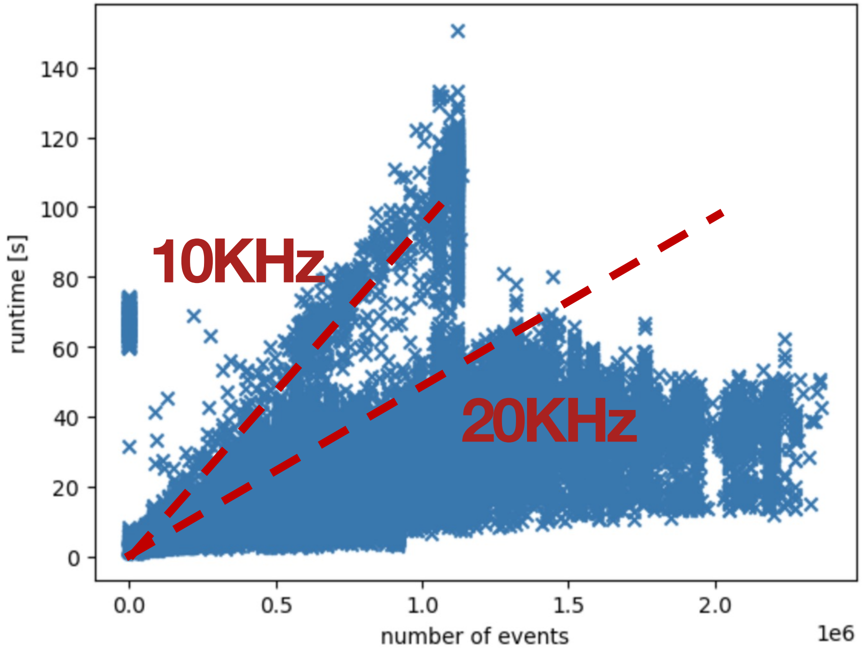


Rates from different, but representative run)

1200 cores across 150 8-core workers

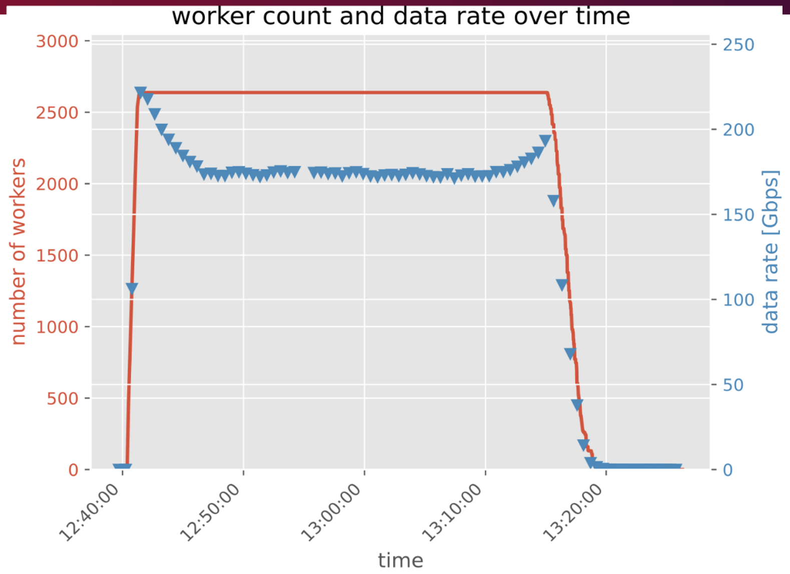


Runtime vs # Events

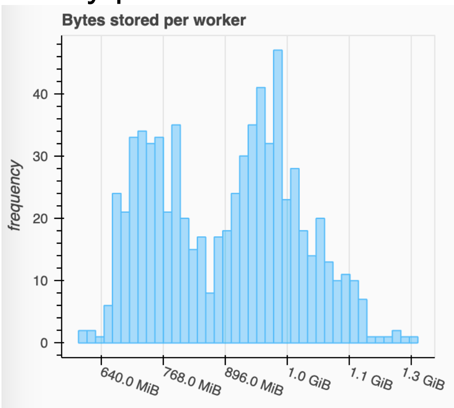


Uproot Toolset, PHYSLITE

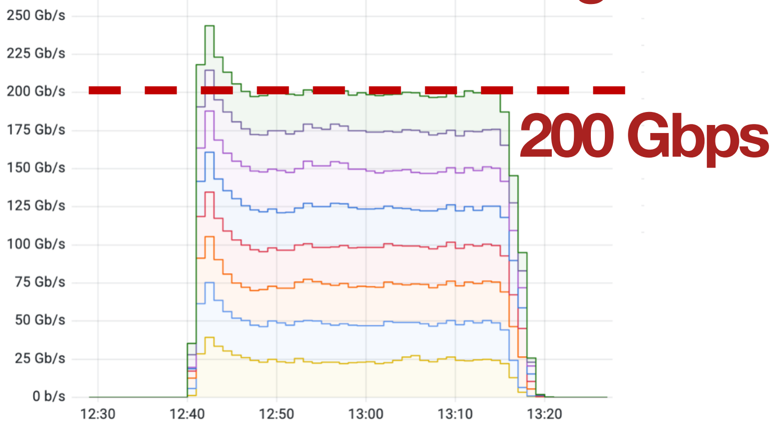
- Several variants were explored at Nebraska; Dask vs TaskVine, dask-jobqueue vs dask-gateway.
- At UChicago, also processed ATLAS PHYSLITE files directly in Python.
  - Goal was using coffea 2024, dask-awkward, uproot; ended up using direct processing in uproot.
  - 218k files, 190TB data, 23B events, ~8kHz/core
- Highlights:
  - Scaled Dask up to around 2.5k cores
  - 200Gbps throughput sustained in network monitoring; slightly less in ‘effective bytes’ into Dask.
- Biggest challenge has been understanding memory usage; significant difference between “uproot only” and the full Coffea 2024.



memory profile across workers

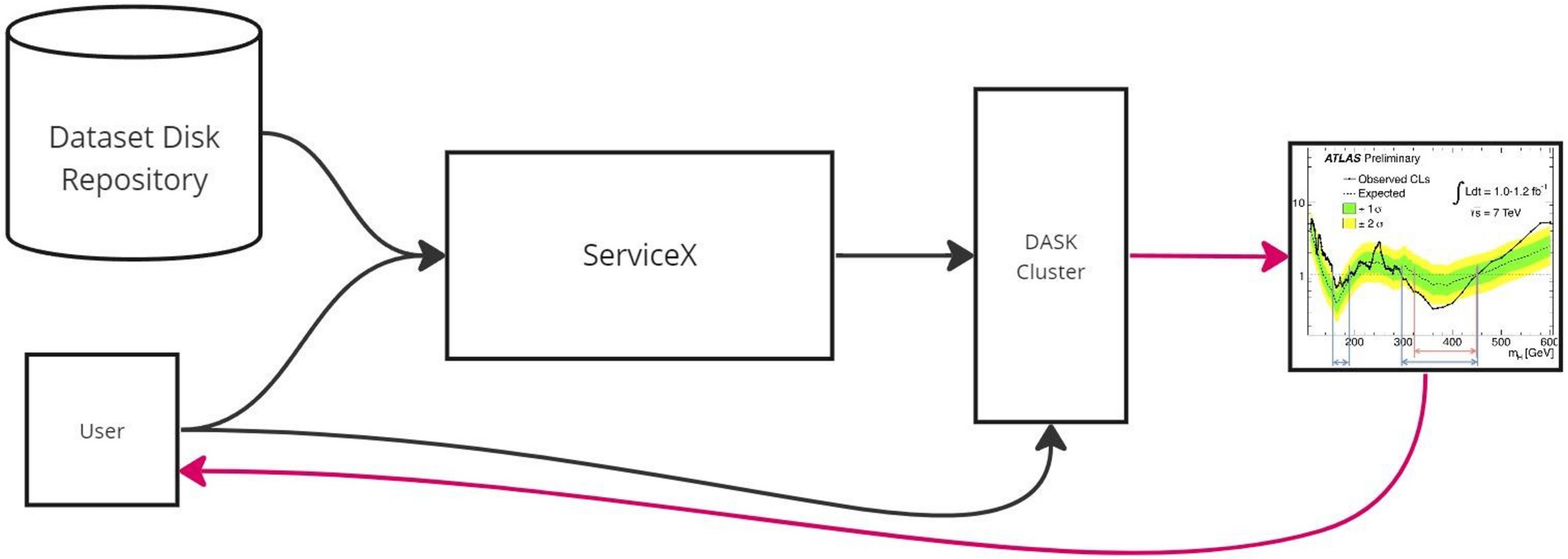


Network monitoring



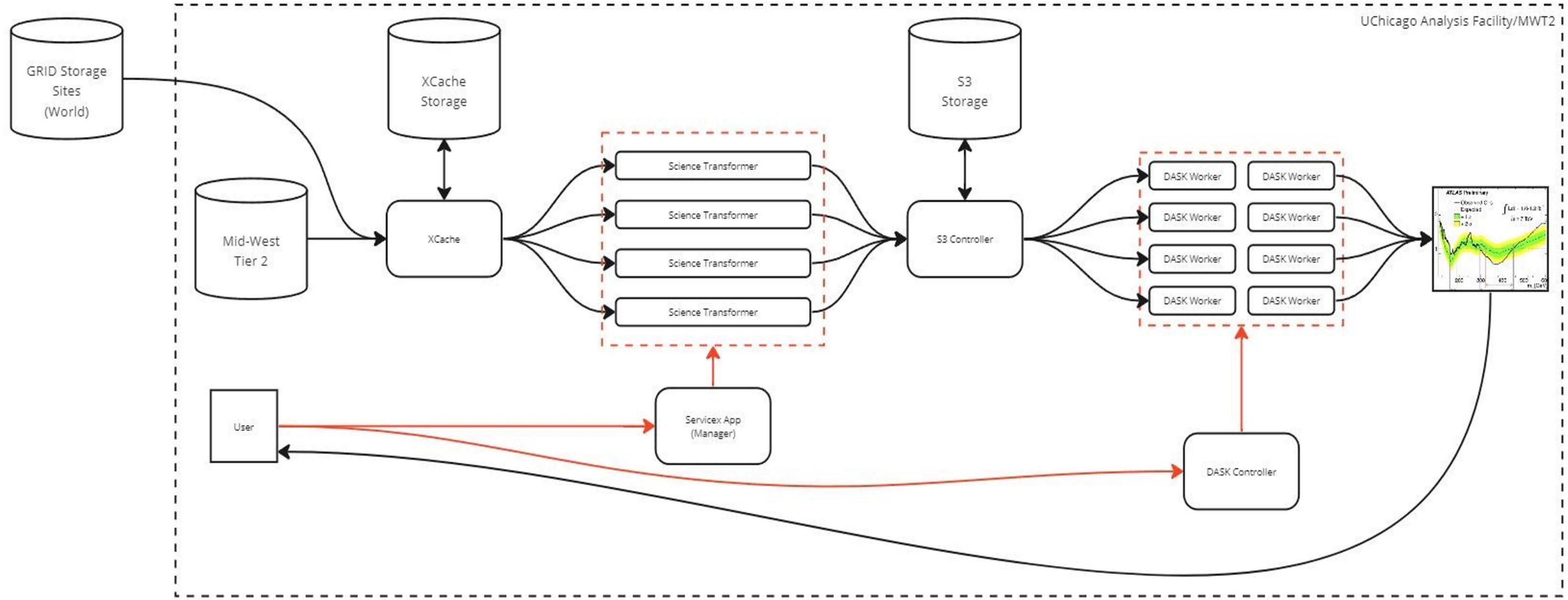
ServiceX Toolset

- ServiceX, developed by IRIS-HEP, derives and delivers columns from datasets via official experiment tools.
  - An ATLAS HL-LHC demonstrator project.
  - This prototype was run at the UChicago facility.
- For the ServiceX toolset, we read data from disk, skimmed with ServiceX, and processed the results with Dask.
  - Goal is the Dask processing step is much quicker and against much smaller dataset
- 230 datasets of ATLAS PHYSLITE data were used to total 200 TB.

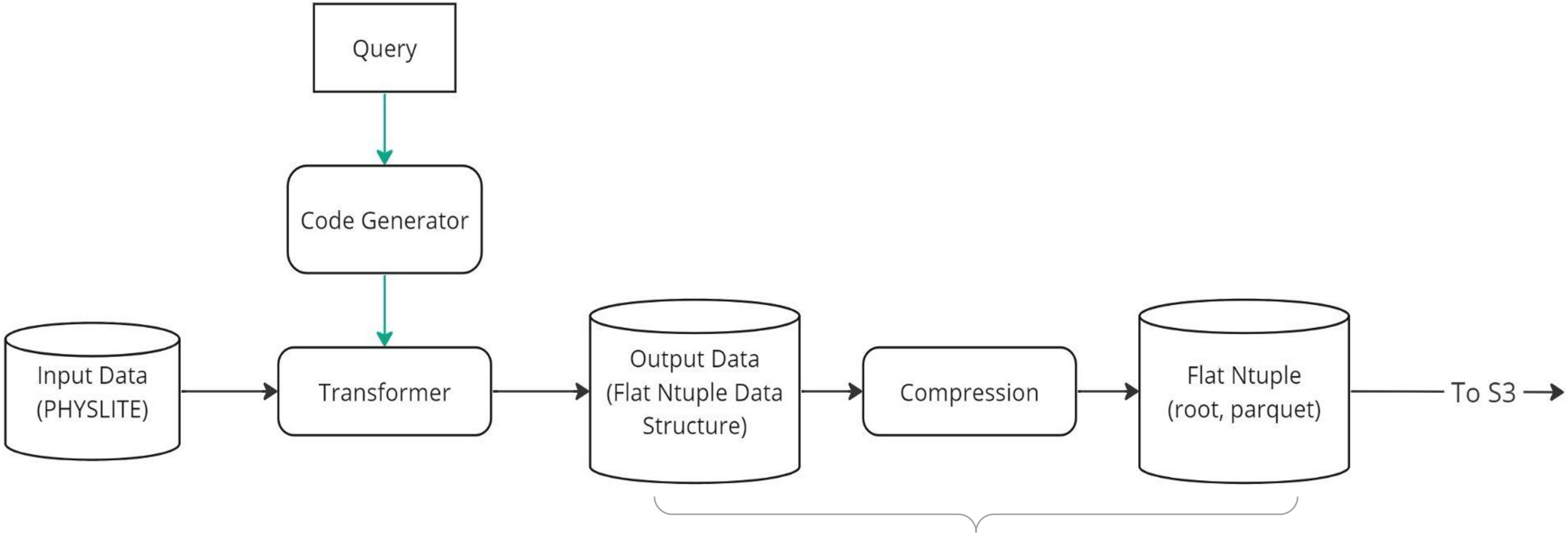


ServiceX Toolset

- XCache was used to cache the PHYSLITE locally and make the storage performance more consistent.
- Between ServiceX and Dask, we stored the temporary ntuples in a local S3 endpoint.
  - The stress put on S3 was one of the main challenges of the ServiceX activities.



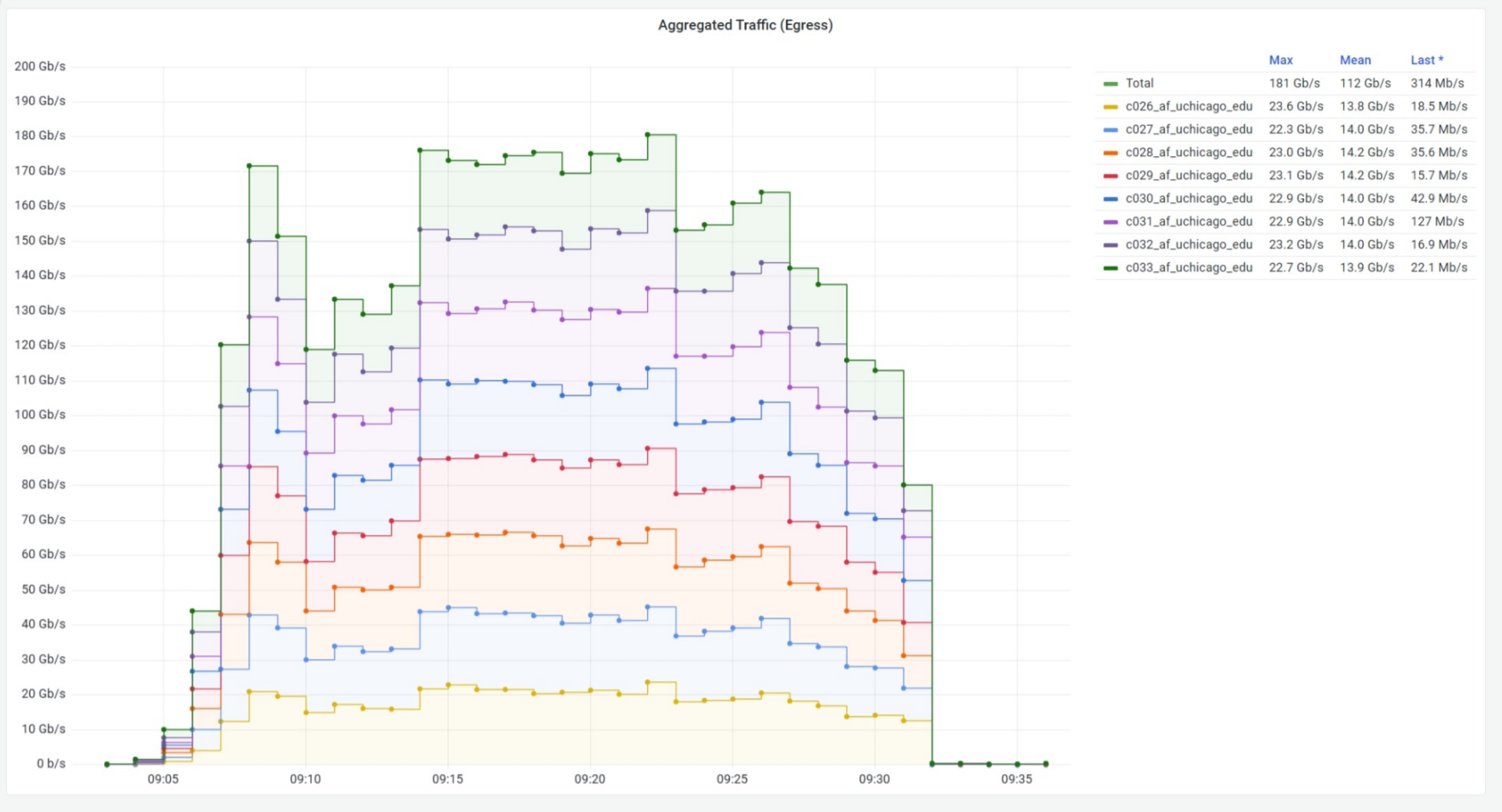
ServiceX up close



Note this intermediate output step wasn't done in the Uproot tests

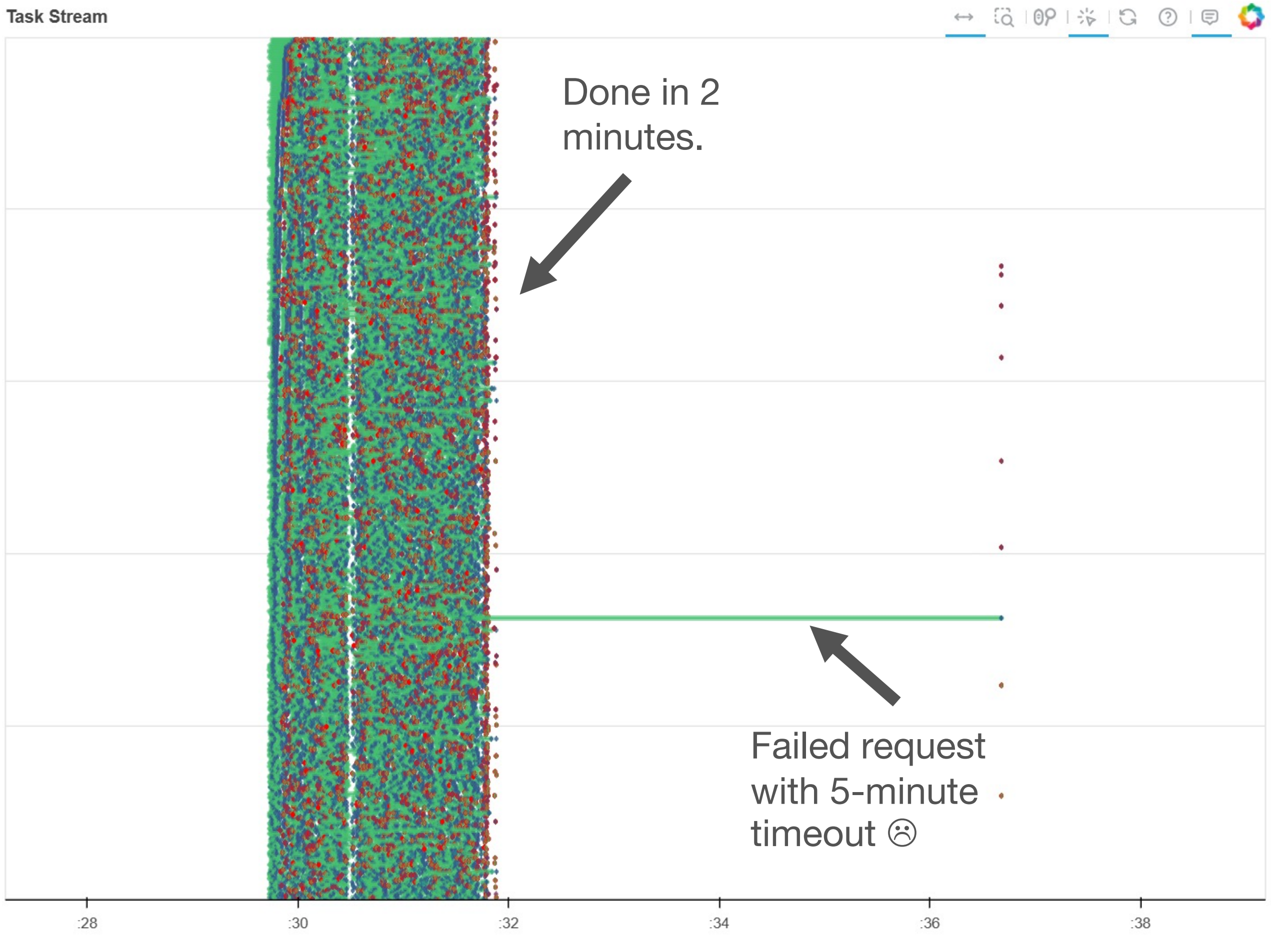
ServiceX Results

- To reduce the overhead of small datasets, we ran on a subset that consisted of the bulk of the data.
- Highlight run:
  - 4 Datasets
  - 146TB total
  - 19,074,862,754 Events
  - 170Gbps
  - Limited to 1,000 pods.
  - Time: 32:28
  - Event Rate: 9,787 kHz



ServiceX Results

- For the Dask step,
  - 500 dask workers
  - Tight skim - around 1 TB of data
  - Skim fraction was 0.5%
  - Event Rate: 198 kHz (due to timeout)
  - Time: 7:20 (5 minutes due to a single timed-out task).



Lessons Learned

FEARLESS SCIENCE

MORGRIDGE
INSTITUTE FOR RESEARCH

Lessons learned

- Python analysis ecosystem:
  - Debugging/**understanding memory usage** is currently the largest challenge. How do you understand memory usage spikes when the behavior is different from your laptop?
    - Nothing unfamiliar here: same applies to C++ code running in HTCondor.
    - Don't forget that Python is a garbage-collected language: GC behavior can have significant impacts.
  - Similarly, the **interaction with storage can be mysterious**: with 100k tasks, strange behaviors that affect 0.01% tasks under load … happens every run.
    - Strange, persistent XRootD errors led to new uproot versions by the end => fixes everyone now benefits from!
- ServiceX:
  - These **at-scale tests have been essential in catching bugs** (missing files when ingesting large datasets, database consistency when stageout to S3 fails, missing retry policies). "Works on my laptop" != "Works in production"
- Facilities:
  - Real, large workflows quickly show network imbalances.
  - Best (better?) practices in **tuning XCache**; scaling achieved is similar to nginx.

Preparing for the HL-LHC

- We have found the "grand challenge" approach to be a useful framing device for focusing effort.
  - A series of increasingly-complex, cumulative exercises towards a common, quantitative goal.
  - This is in addition to the "day to day" effort of bringing projects to fruition.
- Grand Challenges can be both scale and **technology readiness**.
  - Here, we're leaning in technology readiness more.
- We've recently finished an intensive, time-limited exercise to show a vision of analysis at 200Gbps.
  - It's been a resounding success in feeding back issues to developers.
  - We were able to succeed the desired scale at both facilities. **There's nothing about these rates that are out-of-reach**.
    - Facilities were able to identify potential future bottlenecks.
  - In all workflows, we had to sacrifice "realism" in the notebook to get the rates.
    - TODOs around understanding Python ecosystem memory use at this scale.
- Looking to define more realistic & more inclusive challenges in the future.
- Has informed us of "where we are": **now onto the HL-LHC**.

Questions?

This project is supported by the National Science Foundation under Cooperative Agreements OAC-1836650 and PHY-2323298. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

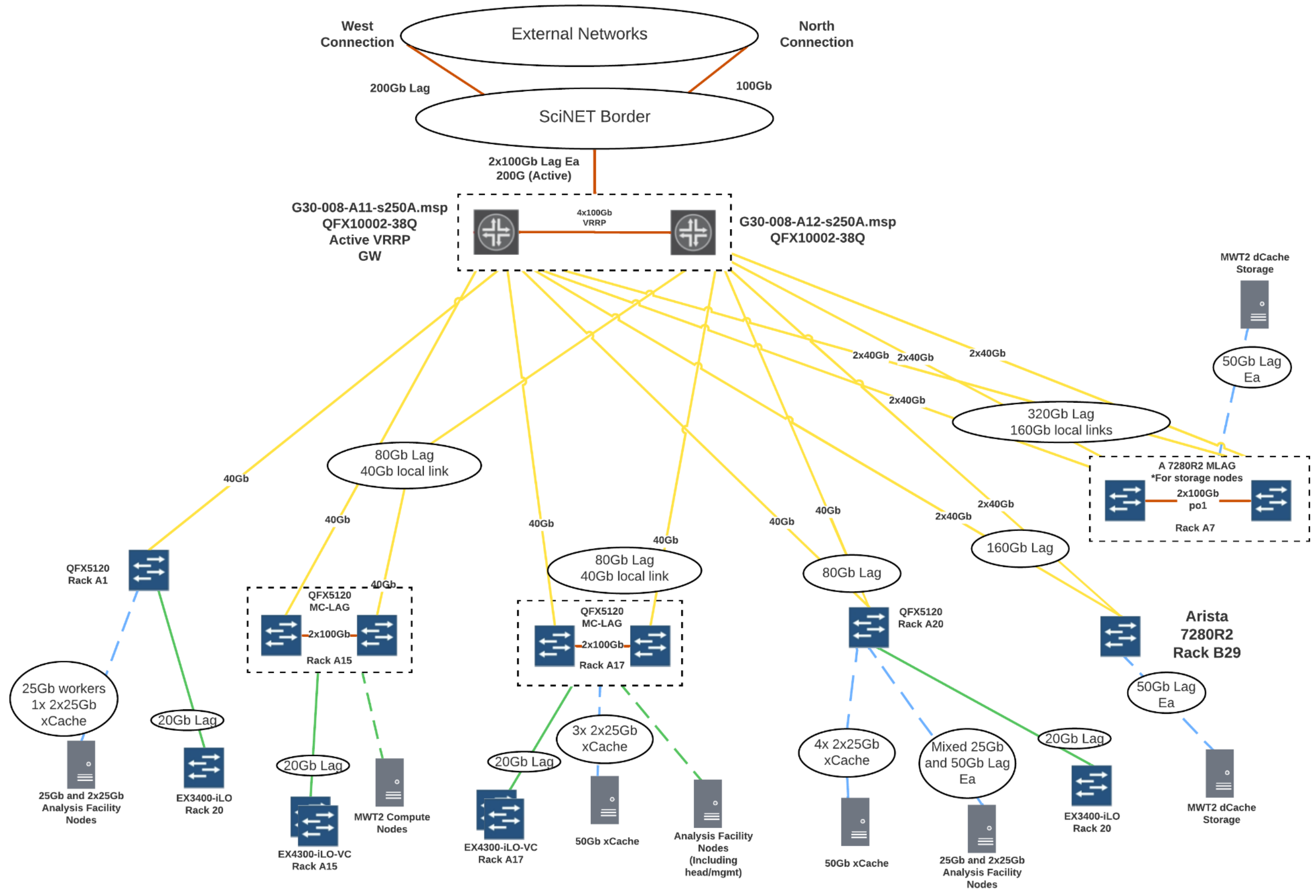
Backup Slides

FEARLESS SCIENCE

MORGRIDGE
INSTITUTE FOR RESEARCH

Chicago Architecture Diagram

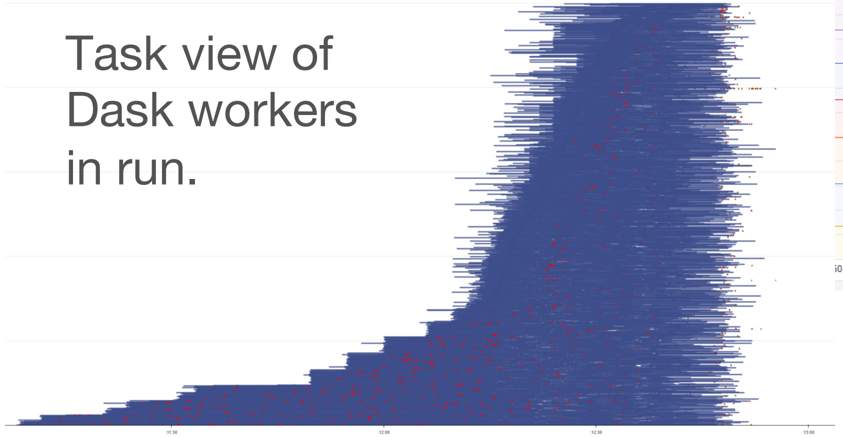
- Compute (transformers/DASK nodes) are in A1, A15, A17, A20.
- All S3 storage nodes are on A20



6045 POD-C	High Speed AF Annex Upgrade	100Gb / 40Gb	25Gb / 10Gb	Created by: MANIAC LAB May 2024	THE UNIVERSITY OF CHICAGO

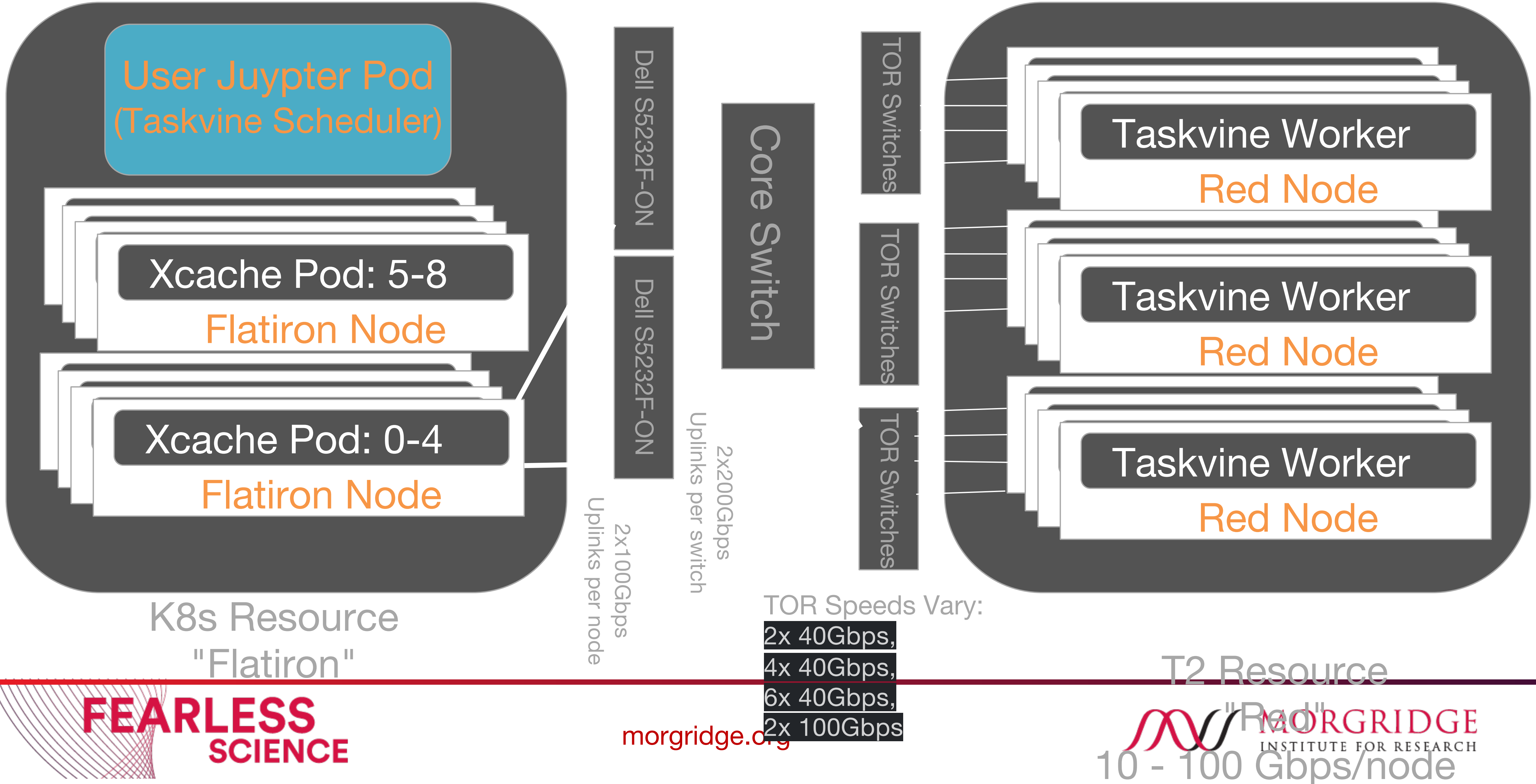
ServiceX Results

- To reduce overhead of small datasets, we focused on a single 50TB dataset.
  - Passed 4 jet events with more than 25 GeV and eta < 2.5.
  - Writes out 2TB of intermediate ntuples.
- Ultimately, was able to achieve 140 gigabits delivered through ServiceX.
- Dask-based processing takes ~2 minutes.

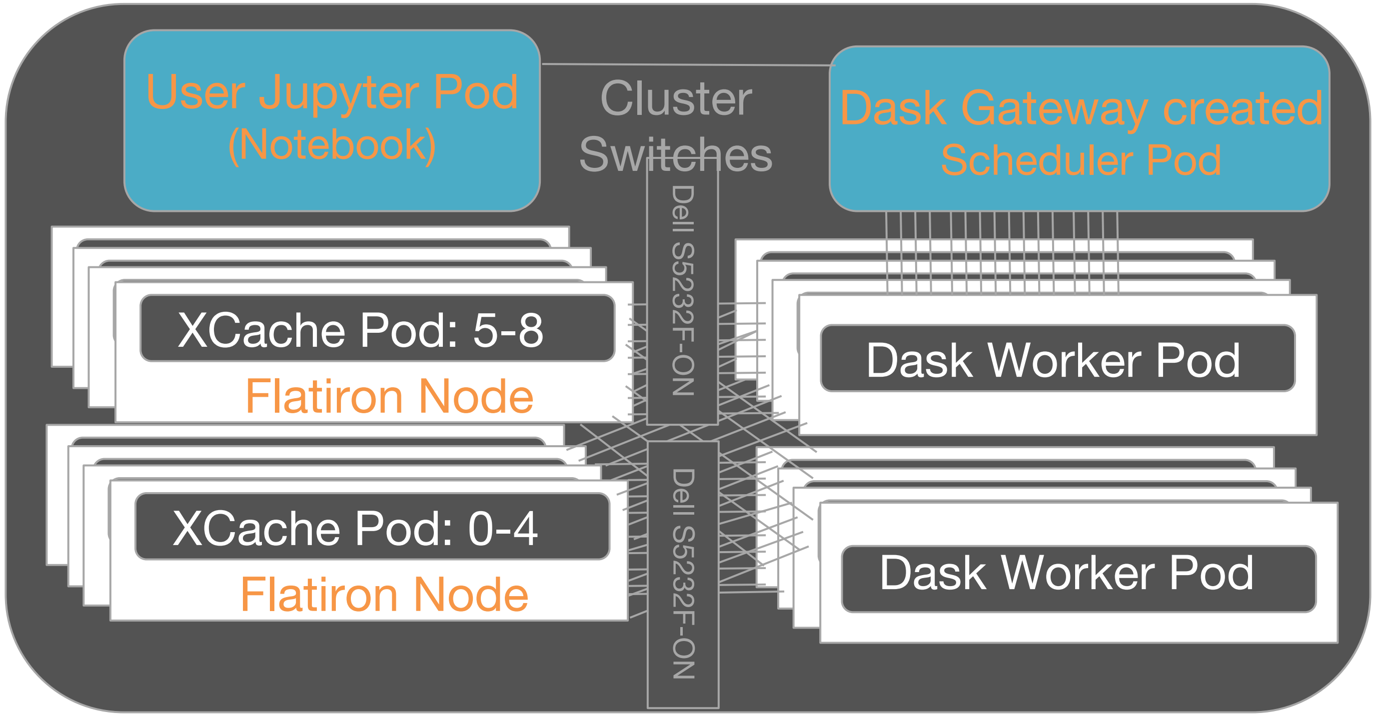


Task view of Dask workers in run.

Nebraska Architecture Diagram



Nebraska Architecture – Dask Gateway



K8s Resource "Flatiron"

2x100Gbps Uplinks per hardware node

XCache Pods given node affinity and local NVME storage (JBOD)