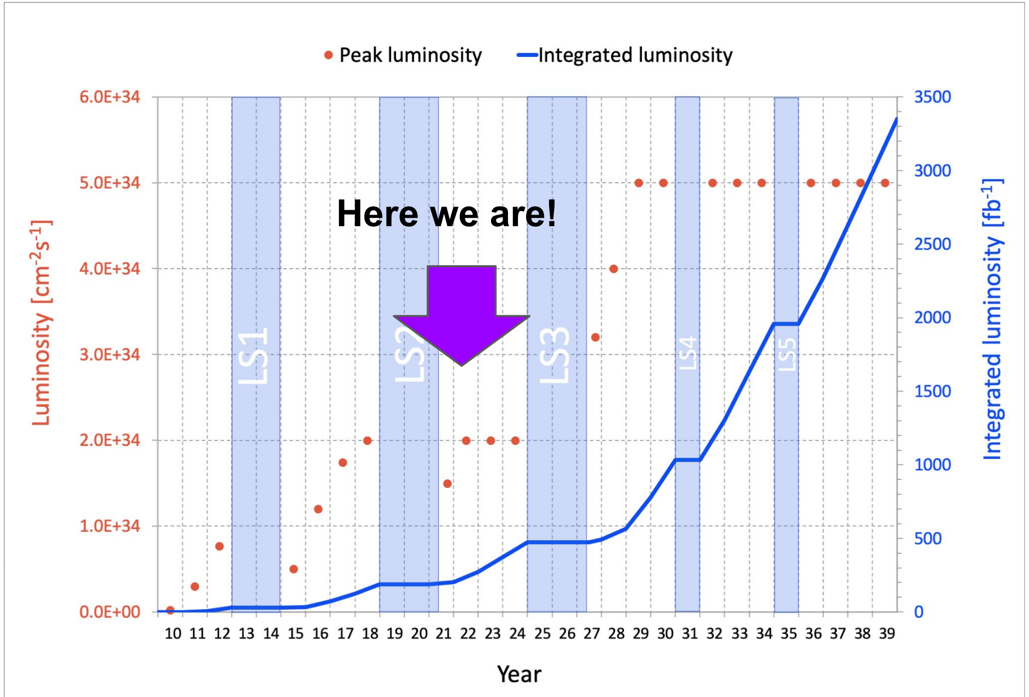
Coffea-casa: an analysis facility prototype

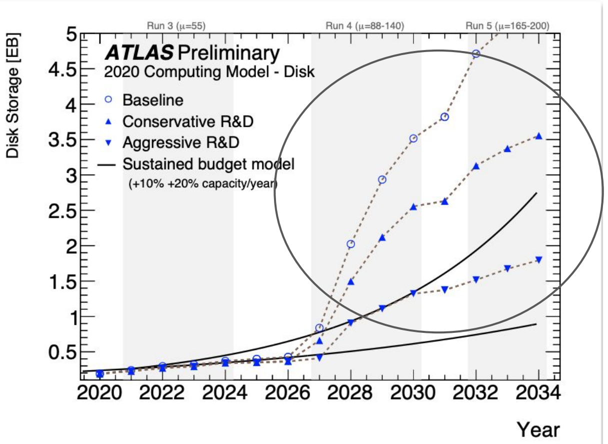
Mat Adamec, Ken Bloom, Oksana Shadura,
University of Nebraska, Lincoln

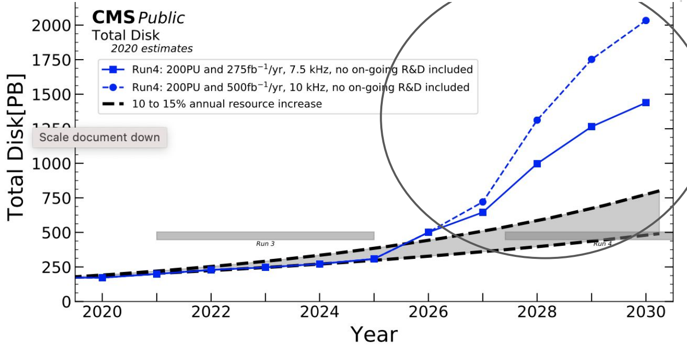
Garhan Attebury, Carl Lundstedt, John Thiltges
University of Nebraska Holland Computing Center

Brian Bockelman
Morgridge Institute

HL-LHC requirements for resources



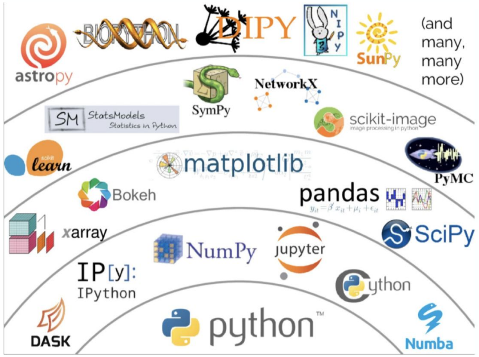




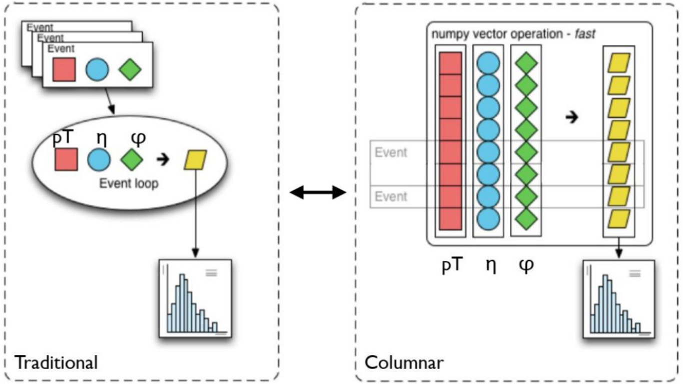
Our goal is driven by the desire *to bring simpler and more agile paradigms for analysis* ***today***, but the scale of the HL-LHC adds more complexity to the existing issue

Looking for new ideas for Analysis Facilities

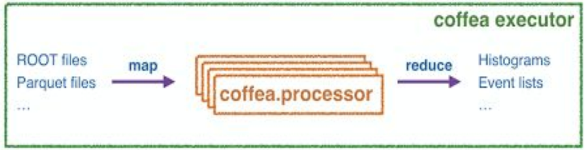
- **New pythonic ecosystem**
- Discovering the benefits of **column-oriented (columnar) data analysis**
- **Interactivity** for user data analysis
- Deliver the needed data to the processing workflow in a fine-grained approach (**data delivery services)** and **efficient storage technologies** (e.g. object stores)
- **Kubernetes (k8s) and** new concept of **"infrastructure as code"**
- **Portability** and flexibility across different environments
- Integration with existing resources: current infrastructure is not going to be replaced in one day



Coffea Analysis Framework



New columnar data analysis concepts!



User just needs to define a high-level wrapper around user analysis code: **the coffea processor** and coffea framework will take care of everything incl. ***scaling-out***

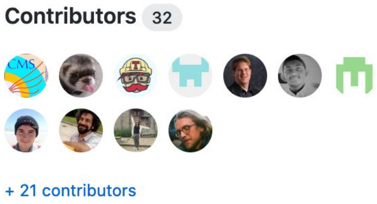
DASK

APACHE Spark

Parsl

Distributed executors!

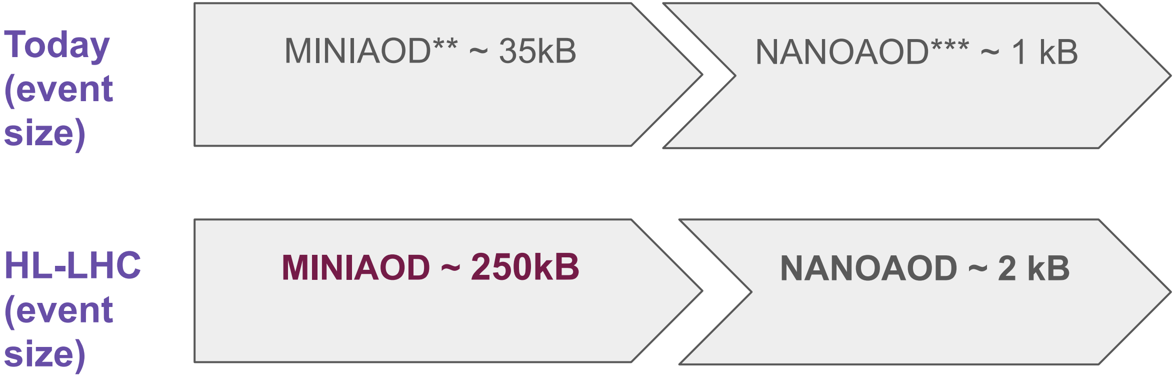
Coffea developers: Lindsey Gray, Matteo Cremonesi, Bo Jayatilaka, Oliver Gutsche, Nick Smith, Allison Hall, Kevin Pedro (**FNAL**); Andrew Melo (Vanderbilt); and others



Contributors 32

+ 21 contributors

HL-LHC Data Challenges*



Today (event size)

MINIAOD** ~ 35kB

NANOAOD*** ~ 1 kB

HL-LHC (event size)

MINIAOD ~ 250kB

NANOAOD ~ 2 kB

If to switch to more compact data format - **“NanoAOD”**, *data volumes should be manageable,* but this format will likely be missing data elements necessary for any given analysis.

Idea: *If we use a sample small enough to be used for interactive analysis in AF: NANOAOD, for example it could be useable as driver for data delivery service to add objects from MINIAOD overnight!*

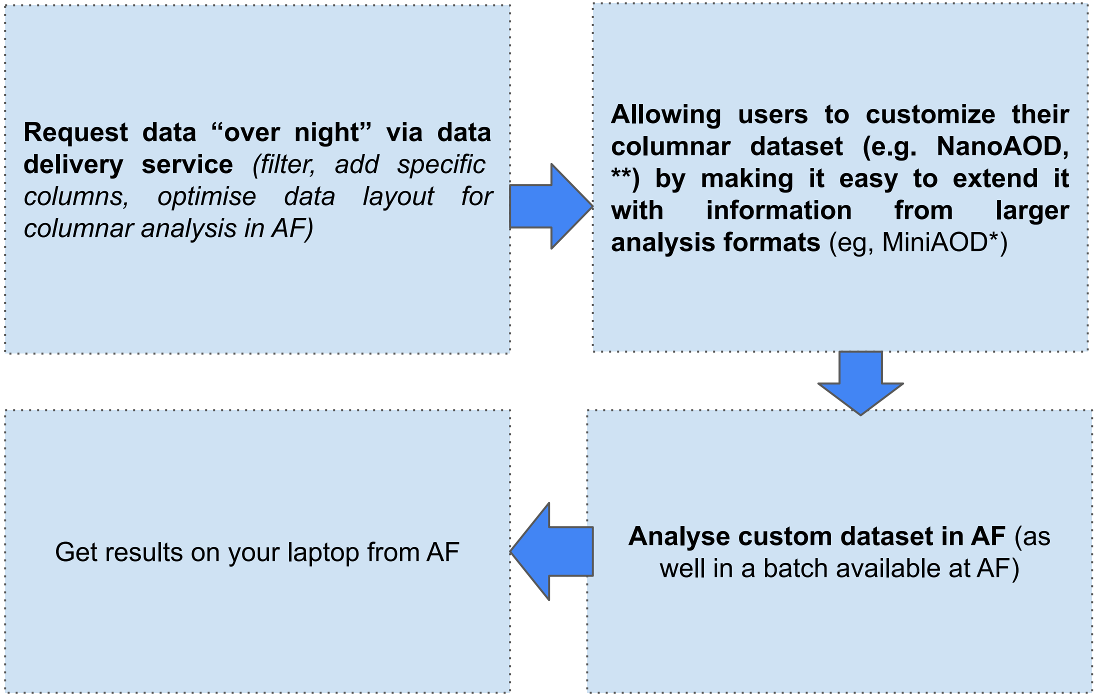
**Here on CMS data formats as an example, shown ideas are applicable for other experiments*

***MiniAOD* - c++ class hierarchy data format

***NanoAOD* - compact, Ntuple like data format readable by bare ROOT

Analysis Facility: “idealistic” user workflow

The AF ***expected to assist with 90% of analyses*** *using NanoAOD** by merging parts or derived from MiniAOD* into NanoAOD** (automatically, without the intervention of the end-user)*



Request data “over night” via data delivery service *(filter, add specific columns, optimise data layout for columnar analysis in AF)*

Allowing users to customize their columnar dataset (e.g. NanoAOD, **) by making it easy to extend it with information from larger analysis formats (eg, MiniAOD*)

Analyse custom dataset in AF (as well in a batch available at AF)

Get results on your laptop from AF

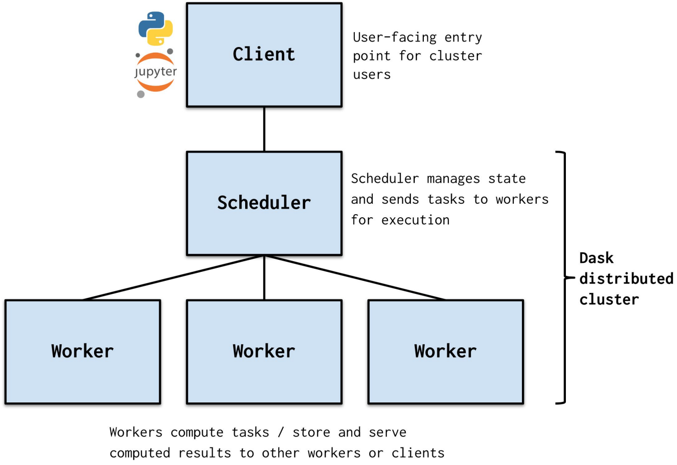
**MiniAOD* - c++ class hierarchy data format

***NanoAOD* - compact, Ntuple like data format readable by bare ROOT

Why we use Dask?

- Dask provides a task-management computational framework in Python based on the manager-worker paradigm
- Dask exposes lower-level APIs letting to build custom systems for in-house applications (!)
- Integrates with HPC clusters, running a variety of schedulers including SLURM, LSF, SGE and *HTCondor via "dask-jobqueue"*
- ***This allows us to create a user-level interactive system via queueing up in the batch system***

Dask can be used inside Jupyter or you can simply launch it through Jupyter and connect directly from your laptop

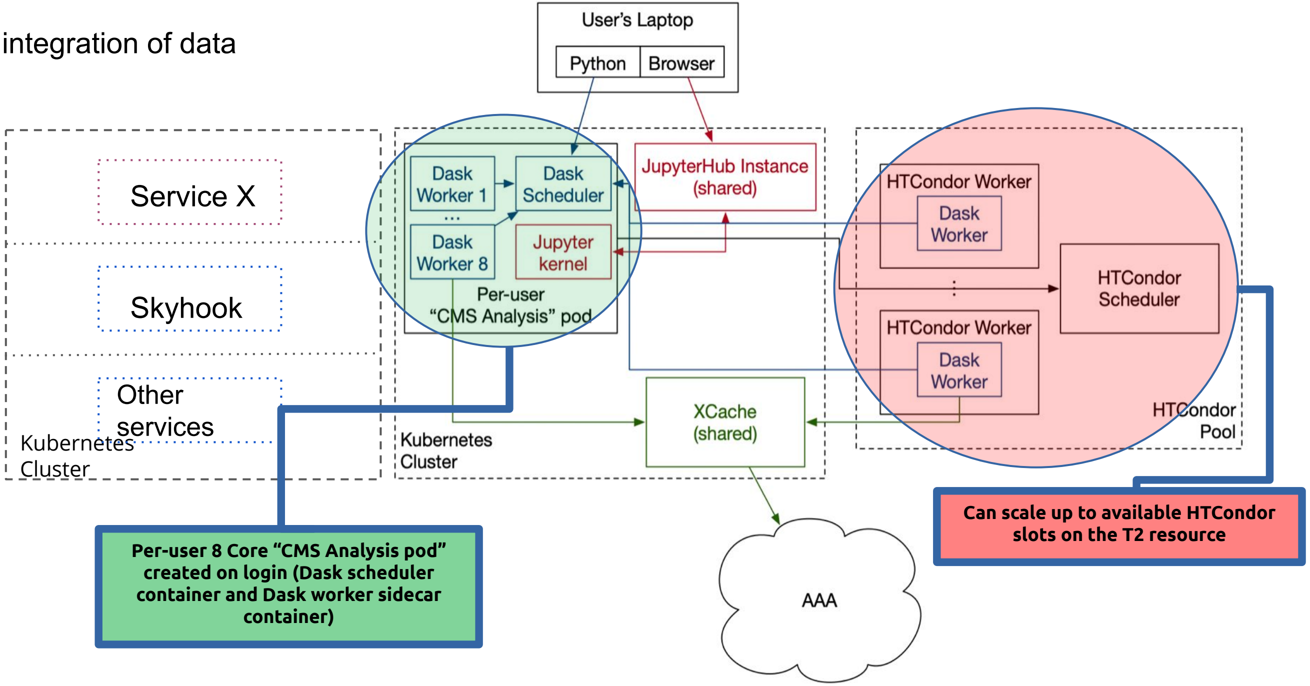


jupyter
Client
User-facing entry point for cluster users
Scheduler
Scheduler manages state and sends tasks to workers for execution
Dask distributed cluster
Worker
Worker
Worker
Workers compute tasks / store and serve computed results to other workers or clients

Analysis Facility Coffea-Casa

We are easily bridging K8s resources with UNL Tier2 resources, while providing interactive environment!

On-going work on integration of data delivery services



Per-user 8 Core "CMS Analysis pod" created on login (Dask scheduler container and Dask worker sidecar container)

Can scale up to available HTCondor slots on the T2 resource

Analysis Facility Coffea-Casa

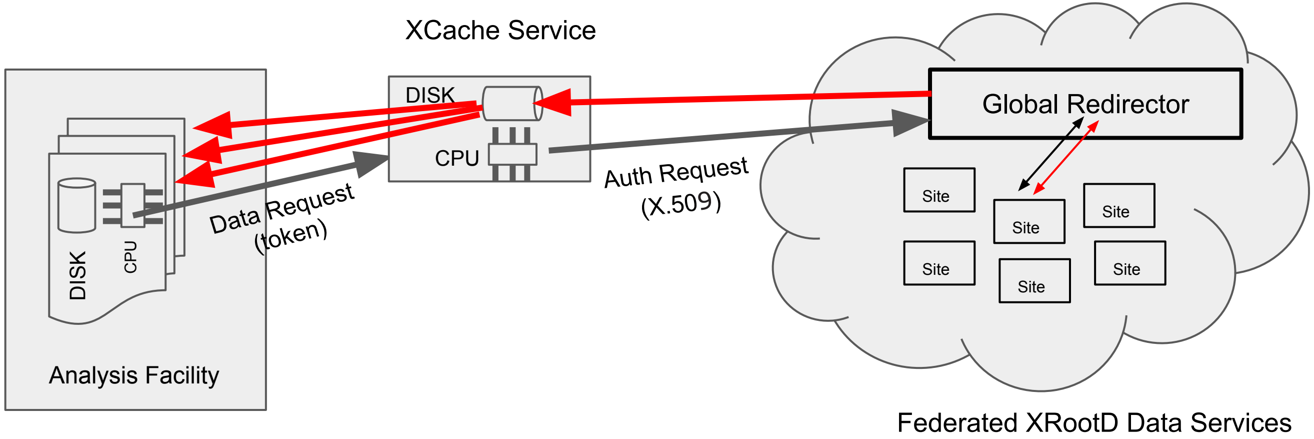
- **Authentication inside the system is independent of grid credentials**
    - Coffea-casa facility uses **OpenID Connect (OIDC)**
    - **Enabled token authentication** for HTCondor:
        - Generated a token for authentication with HTCondor, required for Dask scale-out to the larger resources
    - **Generated a data access token for authentication with a local XRootD server**
    - Generated X.509 credentials (including a CA, host certificate, and user certificate) for use in Dask for TLS as well for user communication to Dask scheduler endpoint
- Security: TLS enabled communication between workers and scheduler by default
- Kubernetes pod customization ‘hook’ to create secrets for services
- Highly customized **“CMS Analysis” Docker container(s)**
- All features are **incorporated into a Helm chart** (Kubernetes packaging format)

Analysis Facility Coffea-Casa

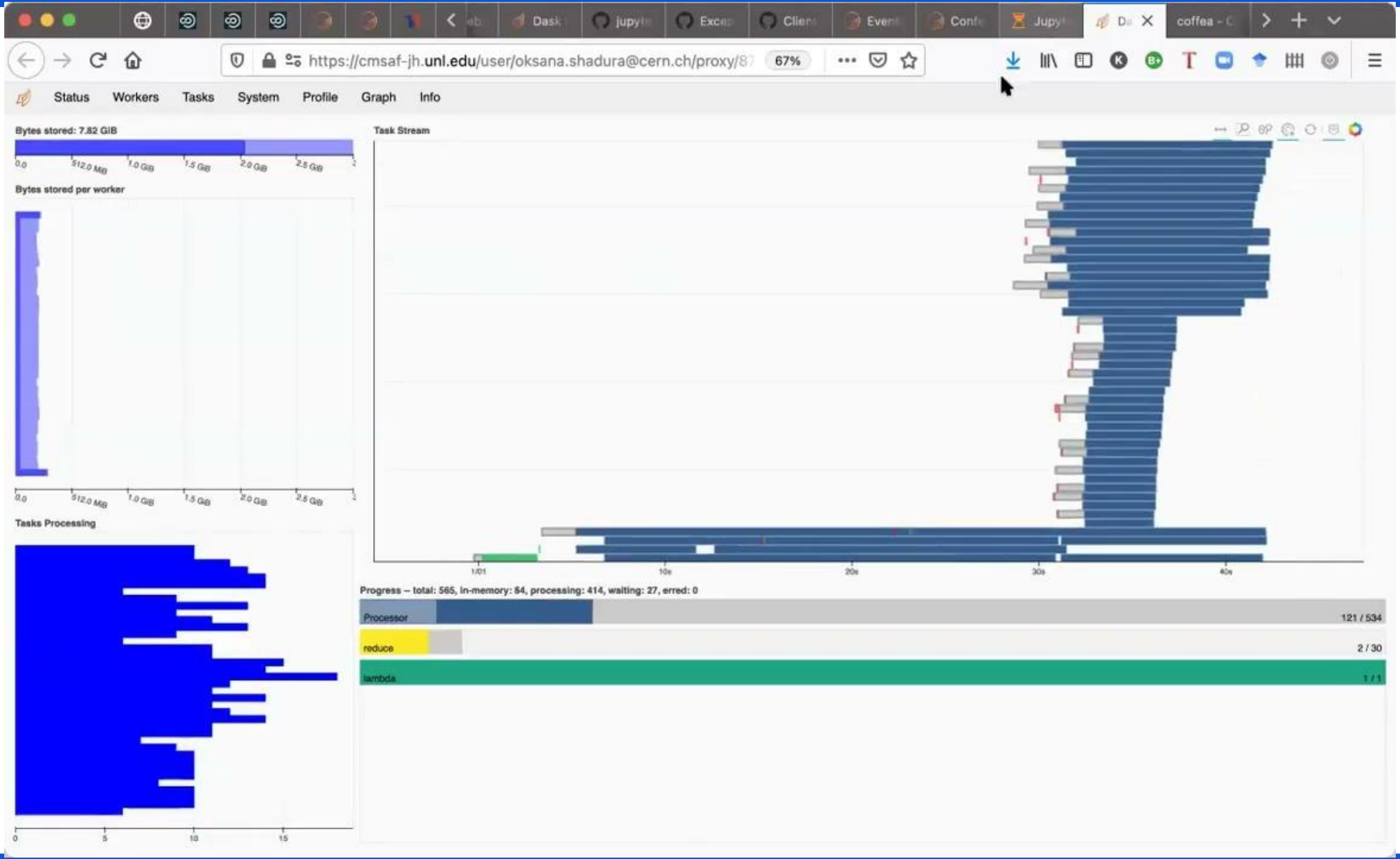
- ***CoffeaCasaCluster:* extending HTCondorCluster integration for Dask**
  - To handle the customizations needed for the Coffea-casa environment, we developed the *CoffeaCasaCluster* object, an extension of ***Dask-jobqueue***'s *HTCondorCluster* object.
  - *CoffeaCasaCluster* ensures the Dask worker starts with the appropriate Docker container in the HTCondor batch system with appropriate configurations and with the firewall ports configured correctly.

Data access @ Analysis Facility

- For speeding up data access Nebraska Tier-2 hosts an **XCache service with 90TB of cache space**
- Access data hosted by an HEP experiment:
    - *no GSI credential within the facility,* **the auto-generated data access token can be used to authenticate with an proxy service based on XRootD/XCache**



Coffea-casa @ UNL - demo



https://cmsaf-jh.unl.edu/user/oksana.shadura@cern.ch/proxy/87

Bytes stored: 7.82 GiB

Progress – total: 565, in-memory: 54, processing: 414, waiting: 27, erred: 0

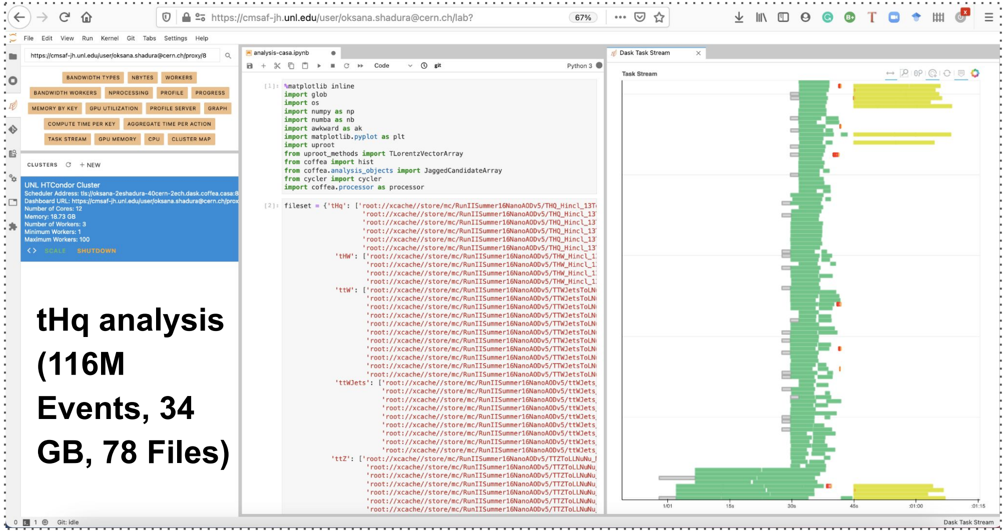
Processor 121 / 534

reduce 2 / 30

lambda 1 / 1

Link to video

Inviting first users



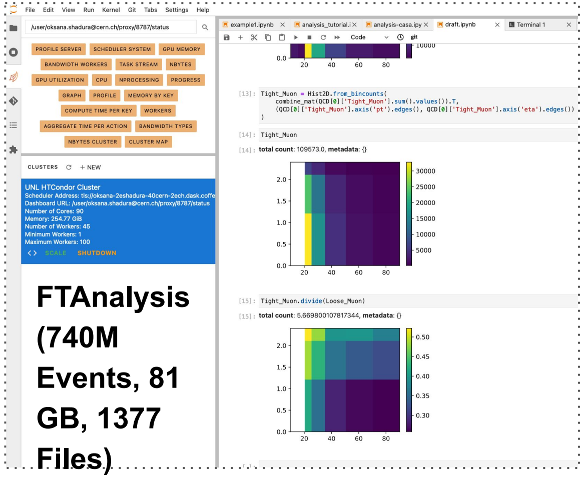
UNL HTCondor Cluster
Scheduler Address: tls://oksana-2eshadura-40cern-2ech.dask.coffea.casa:8
Dashboard URL: https://cmsaf-jh.unl.edu/user/oksana.shadura@cern.ch/prox
Number of Cores: 12
Memory: 18.73 GB
Number of Workers: 3
Minimum Workers: 1
Maximum Workers: 100
SCALE SHUTDOWN

```
[1]: %matplotlib inline
     import glob
     import os
     import numpy as np
     import numba as nb
     import awkward as ak
     import matplotlib.pyplot as plt
     import uproot
     from uproot_methods import TLorentzVectorArray
     from coffea import hist
     from coffea.analysis_objects import JaggedCandidateArray
     from cycler import cycler
     import coffea.processor as processor
```

```
[2]: fileset = {'tHq': ['root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13T
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/THQ_Hincl_13
                 'tHW': ['root://xcache//store/mc/RunIISummer16NanoAODv5/THW_Hincl_1
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/THW_Hincl_1
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/THW_Hincl_1
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/THW_Hincl_1
                 'ttW': ['root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                         'root://xcache//store/mc/RunIISummer16NanoAODv5/TTWJetsToLN
                 'ttWJets': ['root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                             'root://xcache//store/mc/RunIISummer16NanoAODv5/ttWJets_
                'ttZ': ['root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
                        'root://xcache//store/mc/RunIISummer16NanoAODv5/TTZToLLNuNu_
```

tHq analysis (116M Events, 34 GB, 78 Files)

Mat Adamec (UNL undergrad)



UNL HTCondor Cluster
Scheduler Address: tls://oksana-2eshadura-40cern-2ech.dask.coffe
Dashboard URL: /user/oksana.shadura@cern.ch/proxy/8787/status
Number of Cores: 90
Memory: 254.77 GiB
Number of Workers: 45
Minimum Workers: 1
Maximum Workers: 100
SCALE SHUTDOWN

```
[13]: Tight_Muon = Hist2D.from_bincounts(
          combine_mat(QCD[0]['Tight_Muon'].sum().values()).T,
          (QCD[0]['Tight_Muon'].axis('pt').edges(), QCD[0]['Tight_Muon'].axis('eta').edges())
      )

[14]: Tight_Muon

[14]: total count: 109573.0, metadata: {}

[15]: Tight_Muon.divide(Loose_Muon)

[15]: total count: 5.669800107817344, metadata: {}
```

FTAnalysis (740M Events, 81 GB, 1377 Files)

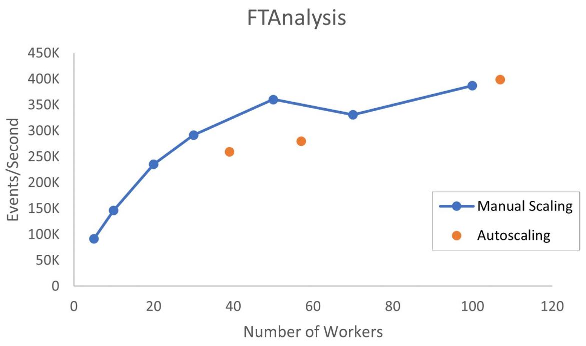
Zora Che (BU CS undergrad)

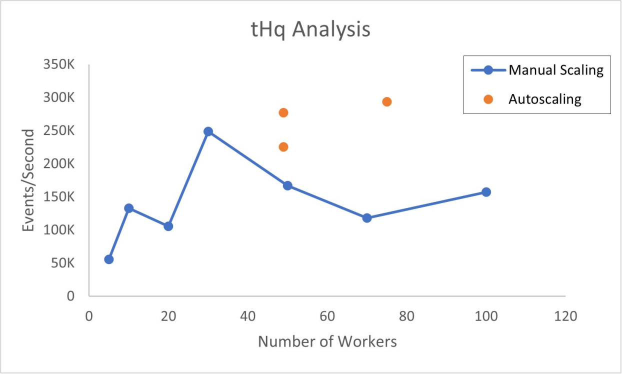
Top quark analyses using the Coffea framework
https://github.com/TopEFT/topcoffea

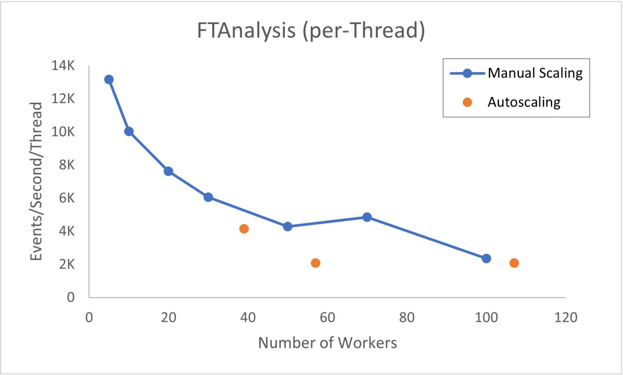
Users already running their analysis on Coffea-casa:

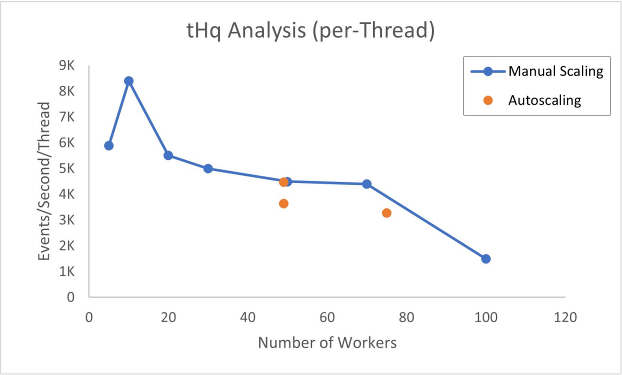
- Some of the examples are **done by undergraduate students**
- **Approachable** (even with all complexity of system behind) and **interactive**

Inviting first users and first performance numbers


FTAnalysis

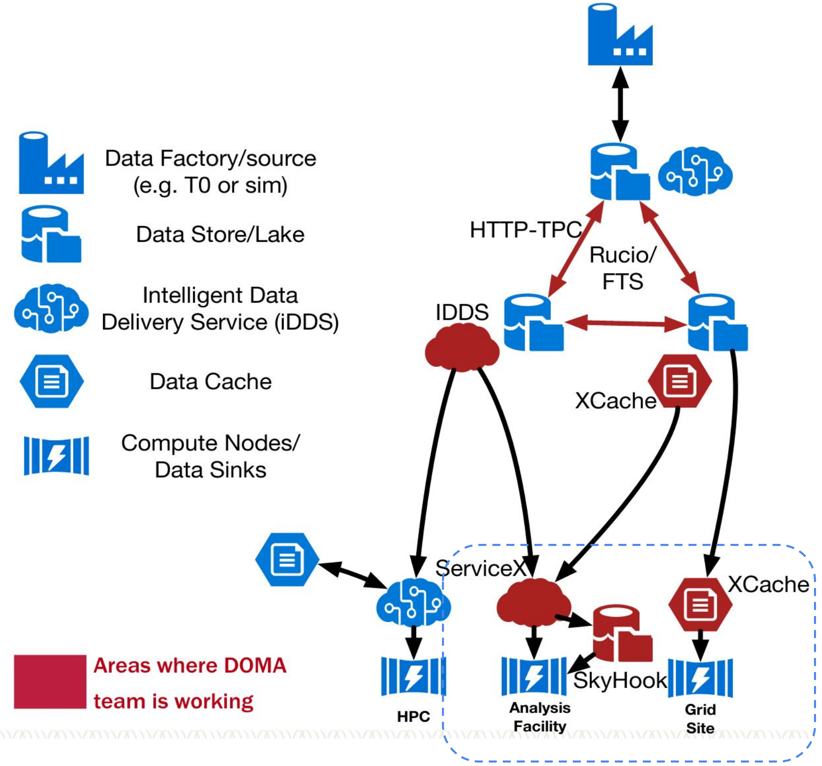

tHq Analysis


FTAnalysis (per-Thread)

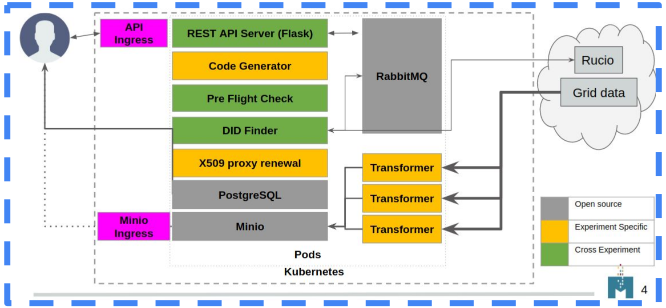

tHq Analysis (per-Thread)

tHq results require some investigation

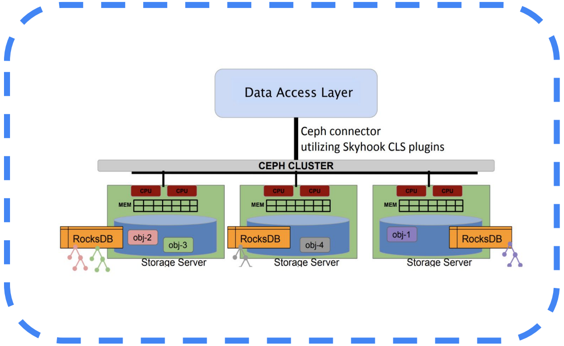
Analysis Facility and Distributed Ecosystem (Data Lakes)



Coffea-casa



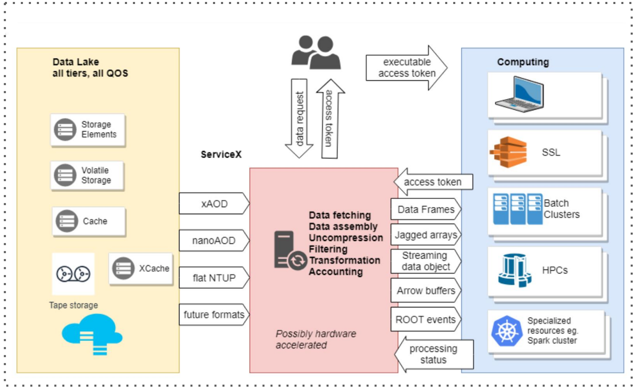
Servicex



Skyhook

Future work: data delivery services @ Coffea-casa

ServiceX

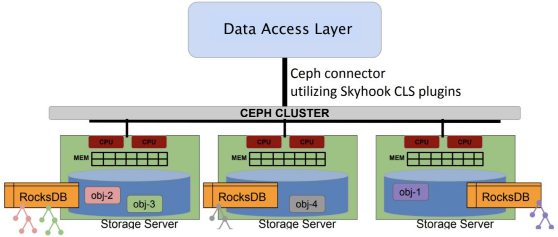


ServiceX provides user level ntuple production

- Converts experiment-specific datasets to columns (e.g. NanoAOD, DAOD)
- Enable simple cuts or simple derived columns and fields (*heavy-weight analysis will still happen via some separate processing toolchain (like CMS CRAB)*

[Check the talk of KyungEon Choi@ vCHEP2021]

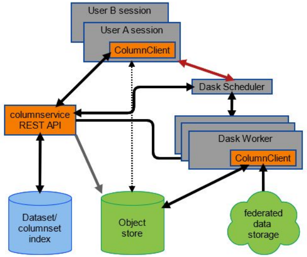
Skyhook DM



The Skyhook DM is converting event data from ROOT files to the internal object-store format

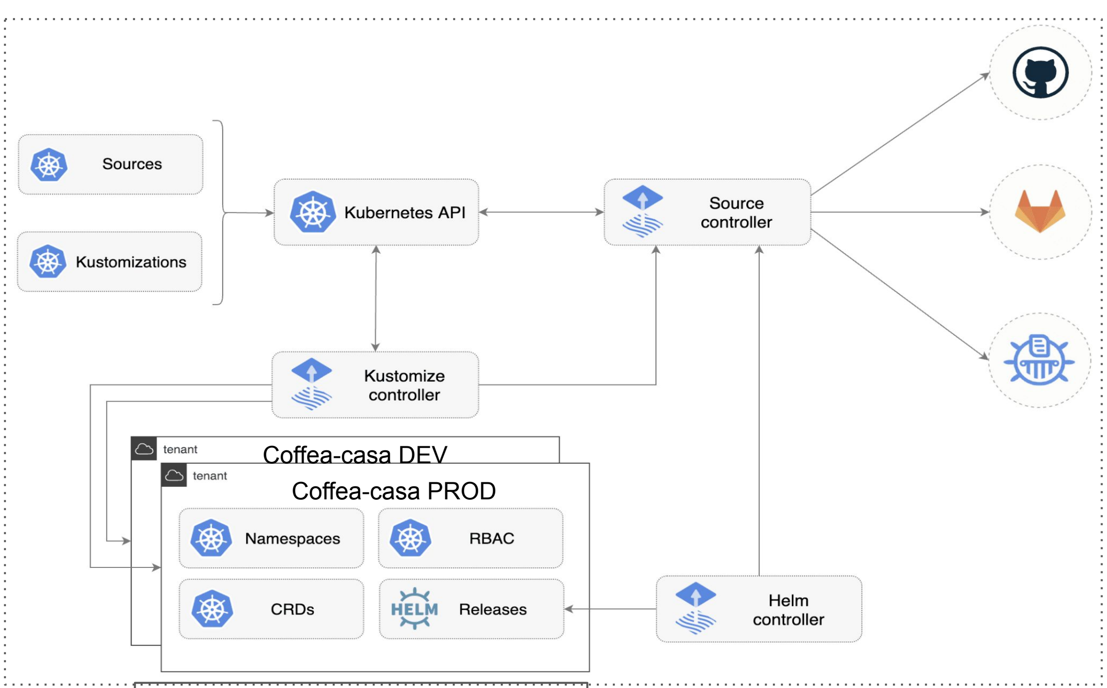
- Provides *extension of Ceph*-side C++ plugins transition from on-disk format to desired memory format
- Uses Dask workers to distribute data to clients
- Data delivered as Arrow tables or via Arrow Dataset API

Columnservice



Columnservice (FNAL) is a multi-tenant service for caching columnar data that removes the need to curate skims and re-run expensive algorithms.

Coffea-casa deployment and distribution: GitOps and multi-tenant Flux



Other tenants: HTCondor, ServiceX,
Skyhook

Docker images will be hosted in **OSG Harbor:** https://hub.opensciencegrid.org

✓ **GitOps** combines ***Git with Kubernetes programmatic service deployment*** properties and serves as an operating model for developing and delivering Kubernetes-based infrastructure and applications.

We expect to package the core infrastructure (e.g., removing the site-specific passwords and secret keys) as a Helm chart**,** which will support different configuration such as **opendata coffea-casa, ATLAS coffea-casa or maybe generic af-casa!**

Coffea-casa deployment: existing coffea-casa AF

- **Two AF facilities** with the possible outcome of adding more sites as soon as we gain experience

CMSAF @T2 Nebraska
“Coffea-casa”
https://cmsaf-jh.unl.edu

Elastic AF @ Fermilab

Developed by: Burt Holzman, Maria Acosta (FNAL)

Conclusions

- The prototype analysis facility at Nebraska, **Coffea-casa, serves as an effective prototype and demonstration of several technologies under development for use in HL-LHC analysis;**
- Coffea-casa demonstrates features such as efficient data access services, notebook interfaces, token authentication, and automatic, external cluster scaling;
- We believe an critical future feature will be access to a **“column service”:** the facility can be used to serve that “column” from a remote site;
- **Initial users have been testing the facility** to provide feedback and the team plans to distribute Coffea-casa products artifacts for use at other Kubernetes-based sites.

This work was supported by the National Science Foundation under Cooperative Agreements OAC-1836650 and PHY-1624356 and awards PHY-1913513 and OAC-1450323.

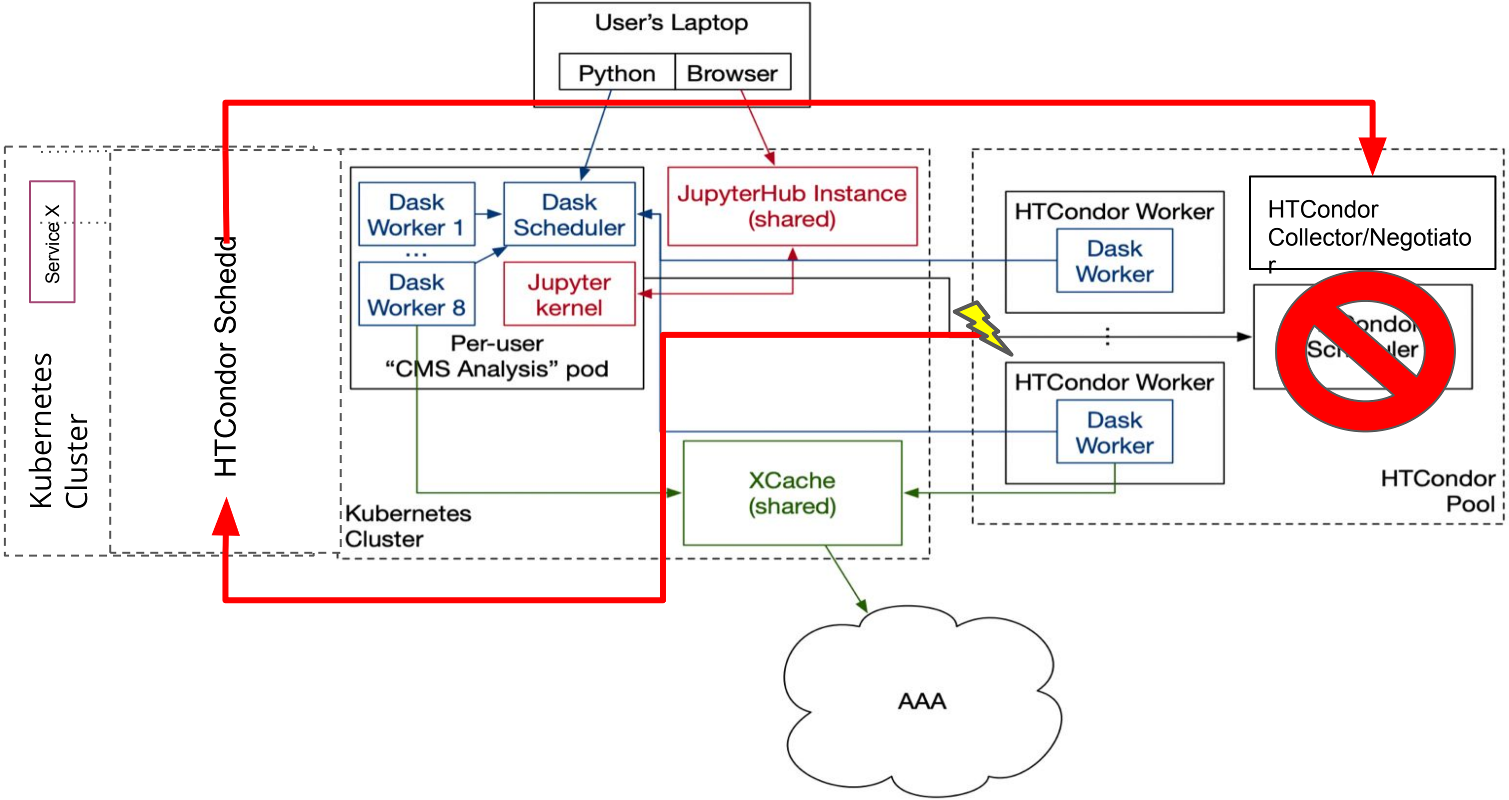
Thank you!

Coffea-casa webpage

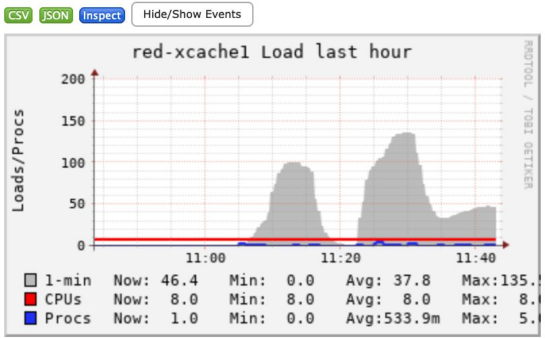
GH discussions

coffea-casa-dev@cern.ch

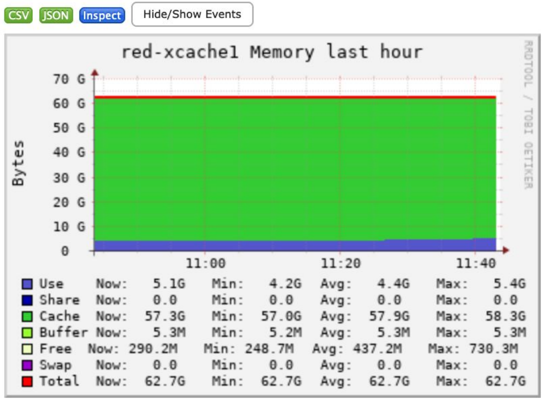
HTCondor: scaling to available Tier2/Tier3 resources



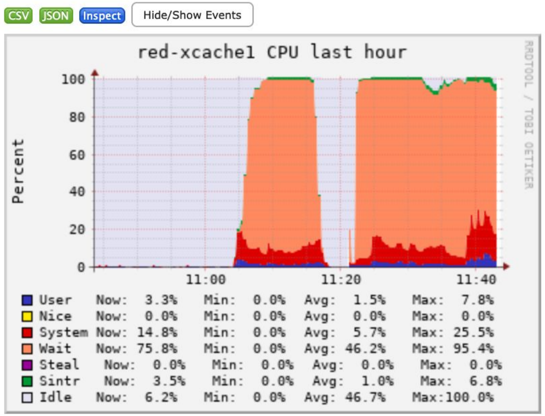
Deploying XCache as a direct dependency Coffea-casa



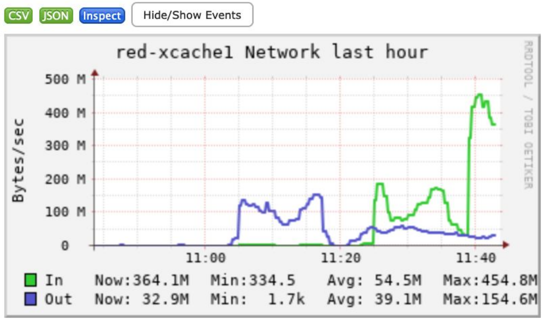
```
■ 1-min   Now: 46.4    Min:  0.0    Avg: 37.8     Max:135.
■ CPUs    Now:  8.0    Min:  8.0    Avg:  8.0     Max:  8.
■ Procs   Now:  1.0    Min:  0.0    Avg:533.9m    Max:  5.
```



```
■ Use    Now:   5.1G    Min:   4.2G   Avg:   4.4G    Max:   5.4G
■ Share  Now:   0.0     Min:   0.0    Avg:   0.0     Max:   0.0
■ Cache  Now:  57.3G    Min:  57.0G   Avg:  57.9G    Max:  58.3G
■ Buffer Now:   5.3M    Min:   5.2M   Avg:   5.3M    Max:   5.3M
□ Free  Now: 290.2M   Min: 248.7M  Avg: 437.2M   Max: 730.3M
■ Swap   Now:   0.0     Min:   0.0    Avg:   0.0     Max:   0.0
■ Total  Now:  62.7G    Min:  62.7G   Avg:  62.7G    Max:  62.7G
```



```
■ User   Now:  3.3%    Min:  0.0%   Avg:  1.5%    Max:  7.8%
■ Nice   Now:  0.0%    Min:  0.0%   Avg:  0.0%    Max:  0.0%
■ System Now: 14.8%    Min:  0.0%   Avg:  5.7%    Max: 25.5%
■ Wait   Now: 75.8%    Min:  0.0%   Avg: 46.2%    Max: 95.4%
■ Steal   Now:  0.0%    Min:  0.0%   Avg:  0.0%    Max:  0.0%
■ Sintr   Now:  3.5%    Min:  0.0%   Avg:  1.0%    Max:  6.8%
□ Idle   Now:  6.2%    Min:  0.0%   Avg: 46.7%    Max:100.0%
```



```
■ In    Now:364.1M  Min:334.5    Avg: 54.5M  Max:454.8M
■ Out   Now: 32.9M  Min:   1.7k  Avg: 39.1M  Max:154.6M
```

- *XRootd file-based caching proxy* (known as Proxy File Cache (PFC) within XRootd code and documentation)
- **Already give visible performance improvement**
- Should be beneficial for datasets caching **in case of skimming**
- We are working on XCache Helm chart for easier deployment