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*
Building blocks used for designing AFs

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modern authentication (AIM/OIDC), tokens, macaroons</td>
</tr>
<tr>
<td>Efficient data delivery and data management technologies</td>
</tr>
<tr>
<td>Columnar analysis and support new pythonic ecosystem</td>
</tr>
<tr>
<td>Modern deployment and integration techniques</td>
</tr>
<tr>
<td>Support for object storage</td>
</tr>
<tr>
<td>Efficient data caching solutions</td>
</tr>
<tr>
<td>Easy integration with existing HPC resources</td>
</tr>
</tbody>
</table>
Building blocks: 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

Coffea developers: Lindsey Gray, Matteo Cremonesi, Bo Jayatilaka, Oliver Gutsche, Nick Smith, Allison Hall, Kevin Pedro (FNAL); Andrew Melo (Vanderbilt); and others
Building blocks: 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.
Grid / cluster site resources

HTCondor scheduler

HTCondor workers

Remote data access

“Interactive” user access node (e.g. LXPLUS, LPC)

Shared resources between users

Simplified diagram of hypothetical Analysis Facility currently used by users
Coffea-casa Interactive Analysis Facility


Grid / cluster site resources

Kubernetes resources

Per-user resources

Shared resources between users

Remote data access

Data flow

X sends requests to Y

JupyterHub
(shared between users)

Dask scheduler

Jupyter kernel

Dask workers

Data delivery services - ServiceX

Skyhook

HTCondor scheduler

HTCondor workers

XCache

Coffea-casa Interactive Analysis Facility


Skyhook
Authentication inside the system is independent of grid credentials

- Coffea-casa facility uses **OpenID Connect (OIDC)**
  - CMS, ATLAS AIM
- **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 **“Analysis” Docker container(s)**
- All features are **incorporated into a Helm chart** (Kubernetes packaging format)
• **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.

• **Looking into new backends:**

  *Workqueue* (http://ccl.cse.nd.edu/software/workqueue/)

• **More features as work in progress**
- 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 components

- **Core**
  - JupyterHub
  - Parallel processors
  - Web-based authentication
  - Dask-scheduler interface
  - Base image(s)

- **Plugins**
  - XCache
  - K8s scaleout
  - HTCondor scaleout
  - BinderHub

- **Batch**
  - ServiceX
  - S3
  - NFS mounts
  - Minio
  - Skyhook

- **Data access**
  - CVMFS

- **Data storage**
  - External authentication
  - workqueue

[Coffea-casa team]
Analysis Facility and Distributed Ecosystem (Data Lakes)
On-going work on integration of data delivery services

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

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
Thank you!

Coffea-casa webpage
GH discussions