Analysis Grand Challenge

Alexander Held (University of Wisconsin–Madison)
Oksana Shadura (University Nebraska–Lincoln)

ACAT 2022: https://indico.cern.ch/event/1106990

This work was supported by the U.S. National Science Foundation (NSF) Cooperative Agreement OAC-1836650 (IRIS-HEP).
AGC: two components

The IRIS-HEP Analysis Grand Challenge (AGC) has two components:

- Defining a **physics analysis task** of realistic HL-LHC scope & scale

- Developing an **analysis pipeline** that implements this task
  - Finding & addressing performance bottlenecks & usability concerns
AGC: how we envisioned it initially

An “integration exercise” for IRIS-HEP

- Demonstrate method for handling HL-LHC data pipeline requirements
  - Large data volumes + bookkeeping
  - Handling of different types of systematic uncertainties
  - Use of reduced data formats (PHYSLITE / NanoAOD), aligned with LHC experiments

- Aiming for “interactive analysis”: turnaround time of ~minutes or less
  - Made possible by highly parallel execution in short bursts, low latency & heavy use of caching

- Specify all analysis details to allow for re-Implementations and re-use for benchmarking

- Execution on Analysis Facilities
AGC: analysis task

Community benchmark

- Analysis task: **ttbar cross-section measurement** in single lepton channel
  - Includes simple top reconstruction
  - Captures relevant workflow aspects and can easily be extended
    - E.g. conversion into a BSM search
  - Analysis task prominently features handling of systematic uncertainties

- Analysis is based on **Run-2 CMS Open Data** (~400 TB of MiniAOD available)
  - Open Data is crucial: everyone can participate
  - Currently using 4 TB of ntupl inputs (pre-converted, ~1B events before cuts)

- Goal of setup is showing **functionality**, not discovering new physics
  - Want to capture workflow; use made-up tools for calibrations & systematic uncertainties
An AGC implementation: software stack

Involves large number of packages from IRIS-HEP and partners

Analysis specific frameworks and packages (available in Docker container)

Data delivery service (k8s)

Optional services (k8s)
Analysis pipeline

- **Pipeline setup**
  - **ServiceX** delivers columns following declarative `func_adl` request
  - **coffea** orchestrates distributed event processing & histogram production
    - Using `uproot`, `awkward-array`, `hist`
  - Visualization with `hist` & `mplhep`
  - Statistical model construction with `cabinetry` & inference with `pyhf`

- **Everything is openly developed** ([IRIS-HEP AGC repository](https://github.com/IRIS-HEP/AGC))
  - Including categorization of datasets in terms of role in AGC demonstrator

- Will be executed on various partner facilities: University Nebraska-Lincoln, UChicago, FNAL, BNL, others

---

**Other AGC implementations:**

- **ROOT RDF** (Andrii Falko, Enrico Guiraud):
  - [andriiknu/RDF/](https://github.com/andriiknu/RDF/)
- **Julia** (Jerry Ling):
  - [Moelf/LHC_AGC.jl](https://github.com/Moelf/LHC_AGC.jl)
Data management tools

Relying on DOMA R&D for fast physics analysis turnaround

Expect **key contributions to improved performance** from three IRIS-HEP DOMA projects:

- **XCache** — XRootd file-based caching proxy used for regional / site caches to store requested on-demand datasets (reducing latency & WAN traffic)
- **ServiceX** — data extraction and delivery delivery service ("column-on-demand" service)
- **Skyhook DM** — an extension of the Ceph distributed storage for the scalable storage of tables and for offloading common data management operations to them (selection, projection, aggregation, and indexing, as well as user-defined functions)
R&D on Analysis Facilities

Rapid prototyping on **coffe-a-casa AF**

- Glueing different areas of IRIS-HEP (AS, SSL & DOMA) together: an AGC execution environment
- Providing environment to explore analysis workflows at scale

Garhan Attebury, Brian Bockelman, Ken Bloom, Carl Lundstedt, John Thiltges, Oksana Shadura, Andrew Wightman
AGC + local files

Behavior in idealistic setup

UChicago Coffea-casa AF @ UChicago ATLAS AF (coffea with FuturesExecutor): reading locally stored files and scaling on local machine (hyperthreading after 48 cores)
AGC: scale-out

Processing 1B events

Columns from flat ntuples processed by coffea

UNL Coffea-casa AF @ UNL CMS Tier-2 (Coffea with DaskExecutor): stable scaling up to 1B events on Tier-2 job queue and efficient scheduling
Results: scaling behavior

I/O and number of cores

UNL Coffea-casa AF @ UNL CMS Tier-2 (Coffea with DaskExecutor): stable scaling to 400 cores

Events with increasing number of branches (bigger fraction of data to read)
ServiceX: Column-on-demand service

- **ServiceX** could bring further performance improvements
  - Initial event filter reduces number of events that need to be processed again
  - Repeated columnar processing can read cached data

- New interesting workflow to be investigated: column addition from parent MiniAOD/PHYS datasets
Extending the analysis task & pipeline

Plans and wishlist

● Expanded analysis task
  ○ Machine learning component in workflow
  ○ Further increased set of systematic uncertainties, more data to process

● Prototype new functionality: on-demand column delivery to enhance information in reduced data formats

● Longer term goal: differentiable analysis pipeline to investigate end-to-end analysis optimization
Conclusions and next steps

- **Promising first performance results** obtained at multiple facilities
  - Execution of full AGC pipeline(s) next summer on various facilities (including diverse hardware configurations)

- Could be used as a baseline benchmark for other communities

- Stay in touch via our mailing list
  - analysis-grand-challenge@iris-hep.org (sign up: Google group)
Backup
AGC: give it a try!
We are making it easy for you to try out our setup

- **One click** to get PyHEP notebook in Binder environment
  - Try it out today!
- You can also use the [UNL Open Data coffea-casa](#)
  - Or [SSL](#) (ATLAS members), or your favorite facility
  - This allows you to scale up (limited on Binder)
  - Everything is available in the [AGC repository](#)