Physics analysis workflows and pipelines for the HL-LHC

Alexander Held[1], Oksana Shadura[2]

[1] University of Wisconsin–Madison

[2] University of Nebraska–Lincoln

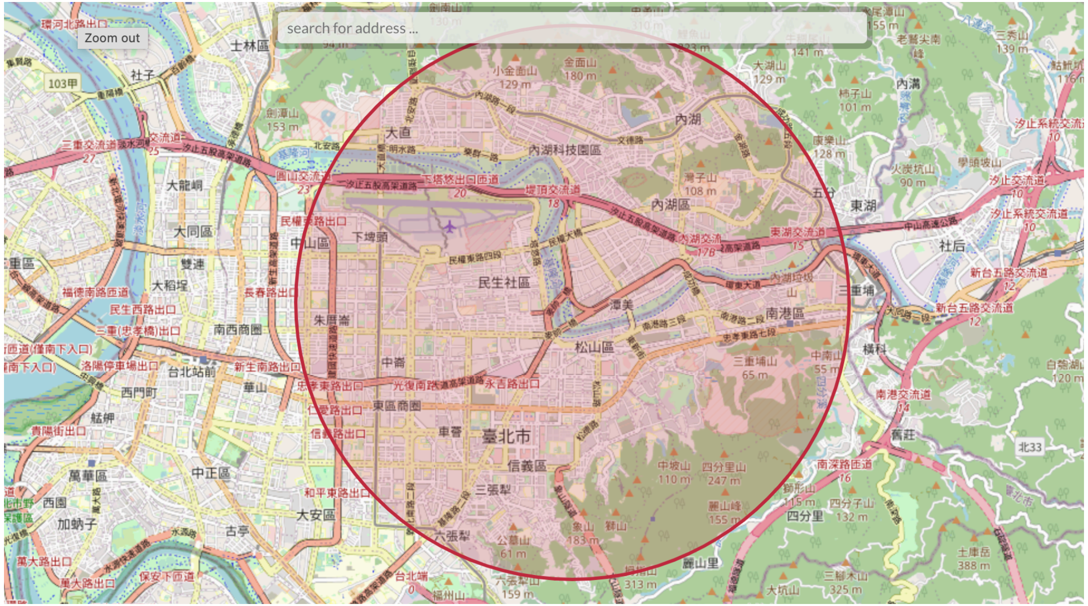
ISGC 2023
https://indico4.twgrid.org/event/25/
March 23, 2023

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

The Large Hadron Collider

- The **Large Hadron Collider (LHC)** collides protons with 13.6 TeV in a 27 km tunnel, 100 m underground
- **Collisions recorded** by multiple detectors and (after a lot of processing) available to physicists as **columnar data**
    - 1 row per collision event, filled with nested data characterizing collision

LHC ring overlaid over Taipei

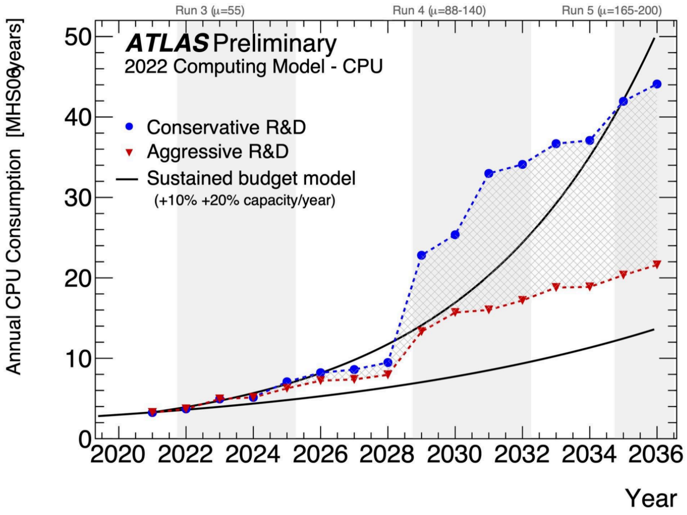


https://natronics.github.io/science-hack-day-2014/lhc-map/

Data analysis at the LHC and the HL-LHC

- Focusing here on the **final steps of data analysis**
    - physicists turning columnar data into results ready for publication
        - input: nested data structure per collision (~billions of rows)
        - output: results of statistical inference, figures, tables, …

- The upcoming **High Luminosity LHC** poses **computational challenges**
    - significant data volume increases
    - R&D required to scale to the data analysis demands



CERN-LHCC-2022-005

projected computing resources required for ATLAS (not limited to end-user data analysis)

The Analysis Grand Challenge (AGC)

- The **“Analysis Grand Challenge” (AGC)** aims to help **address the computing challenges** of the HL-LHC

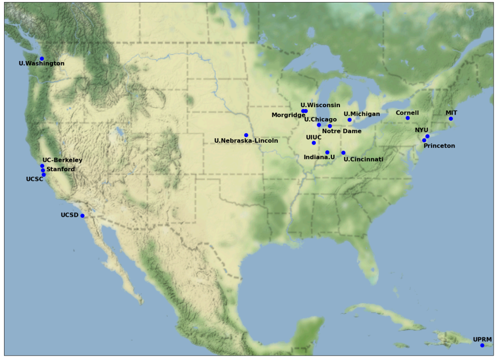
- The AGC has **two components**

  1. define a physics analysis task of realistic scope & scale
  2. develop an analysis pipeline that implements the task

     - find & address performance bottlenecks & usability concerns

IRIS-HEP and the Analysis Grand Challenge

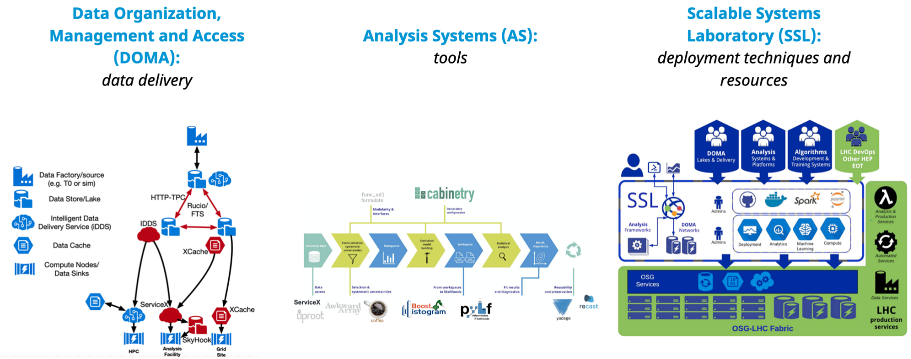
- **IRIS-HEP**: *"Institute for Research and Innovation in Software for High Energy Physics"*
  - software institute funded by the US National Science Foundation
  - research & development for the HL-LHC
    - innovative algorithms for data reconstruction & triggering
    - analysis systems to reduce time-to-insight and maximize physics potential
    - data organization, management and access systems
  - more information: https://iris-hep.org/



institutes participating in IRIS-HEP

IRIS-HEP and the Analysis Grand Challenge

- **AGC**: *“Analysis Grand Challenge”*
  - historically, an integration exercise
    - test realistic end-to-end analysis pipelines aimed at HL-LHC use
    - combine technologies being developed in various ares of IRIS-HEP & adjacent ecosystem
    - identify & address performance bottlenecks and usability issues
  - organized jointly with the US ATLAS & US CMS operations programs

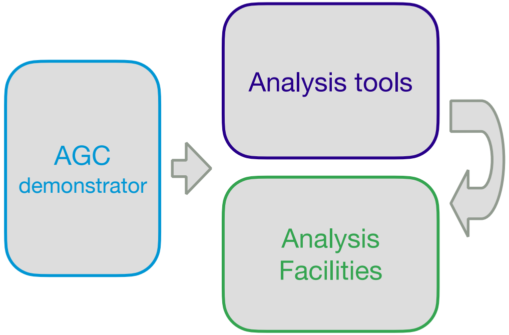


Data Organization, Management and Access (DOMA):
data delivery

Analysis Systems (AS):
tools

Scalable Systems Laboratory (SSL):
deployment techniques and resources

AGC combining IRIS-HEP focus areas



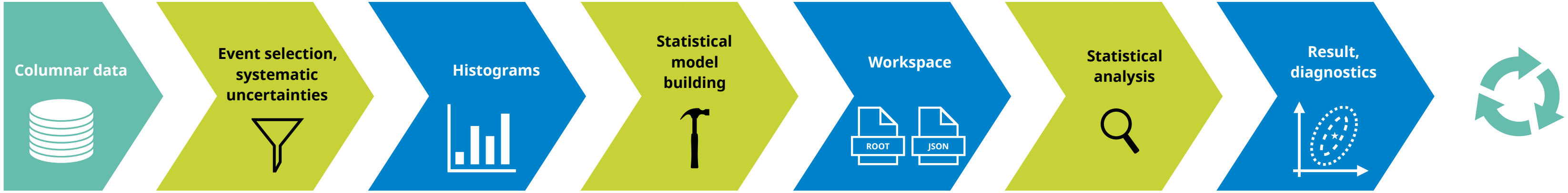
AGC demonstrator

Analysis tools

Analysis Facilities

“Analysis” in the AGC context

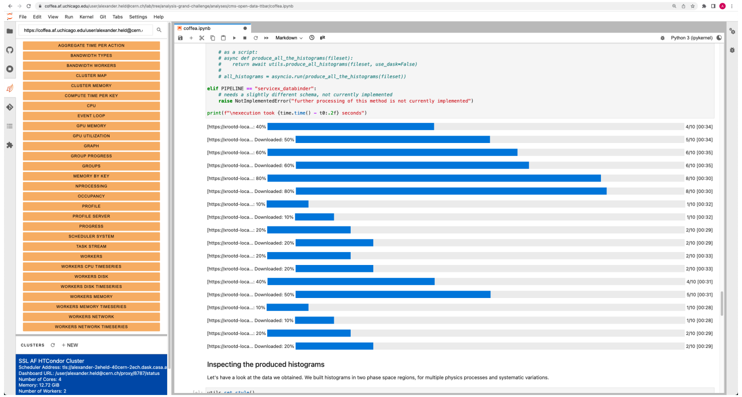
- In view of the HL-LHC: “analysis” **starts** from centrally produced **common data samples** (= big tables of information)
- Includes all **subsequent steps** to produce results needed for publication
  - extract relevant data
  - (re-) calibrate objects (groups of columns) & calculate systematic variations (new columns)
  - filter events (rows) & calculate observables (new columns)
  - histogramming (for binned analyses)
  - construct statistical model + perform statistical inference
  - visualize results & provide all relevant information to study analysis details
- Do all these steps in a **reproducible** way



Moving beyond an integration exercise

- Investigating the possibility of **“interactive analysis”**: turnaround time of minutes or less
  - made possible by highly parallel execution in short bursts, low latency & heavy use of caching
- We hope that the AGC can be **useful to the broader community**!
  - testbed for software library development
  - environment to prototype analysis workflows
  - functionality & integration test for analysis facility development

interactive analysis in a notebook



SSL AF HTCondor Cluster
Scheduler Address: tls://alexander-2eheld-40cern-2ech.dask.casa.a
Dashboard URL: /user/alexander.held@cern.ch/proxy/8787/status
Number of Cores: 4
Memory: 12.72 GiB
Number of Workers: 2

```python
    # as a script:
    # async def produce_all_the_histograms(fileset):
    #     return await utils.produce_all_histograms(fileset, use_dask=False)
    #
    # all_histograms = asyncio.run(produce_all_the_histograms(fileset))

elif PIPELINE == "servicex_databinder":
    # needs a slightly different schema, not currently implemented
    raise NotImplementedError("further processing of this method is not currently implemented")

print(f"\nexecution took {time.time() - t0:.2f} seconds")
```

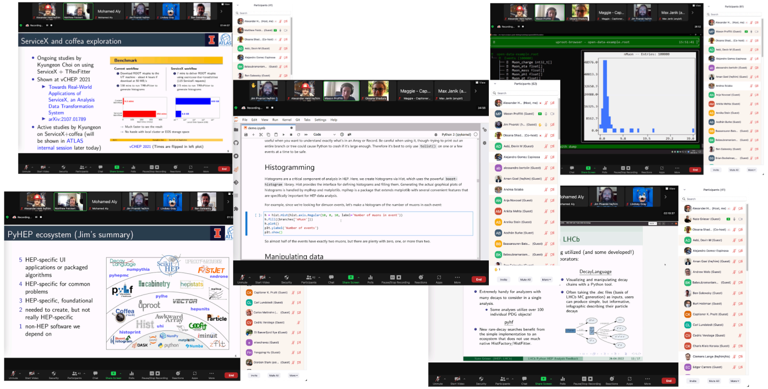
```
[https://xrootd-loca...: 40%                4/10 [00:34]
[https://xrootd-loca... Downloaded: 50%     5/10 [00:34]
[https://xrootd-loca...: 60%                6/10 [00:35]
[https://xrootd-loca... Downloaded: 60%     6/10 [00:35]
[https://xrootd-loca...: 80%                8/10 [00:30]
[https://xrootd-loca... Downloaded: 80%     8/10 [00:30]
[https://xrootd-loca...: 10%                1/10 [00:32]
[https://xrootd-loca... Downloaded: 10%     1/10 [00:32]
[https://xrootd-loca...: 20%                2/10 [00:29]
[https://xrootd-loca... Downloaded: 20%     2/10 [00:29]
[https://xrootd-loca...: 20%                2/10 [00:33]
[https://xrootd-loca... Downloaded: 20%     2/10 [00:33]
[https://xrootd-loca...: 40%                4/10 [00:31]
[https://xrootd-loca... Downloaded: 50%     5/10 [00:31]
[https://xrootd-loca...: 10%                1/10 [00:28]
[https://xrootd-loca... Downloaded: 10%     1/10 [00:28]
[https://xrootd-loca...: 20%                2/10 [00:29]
[https://xrootd-loca... Downloaded: 20%     2/10 [00:29]
```

Inspecting the produced histograms

Let's have a look at the data we obtained. We built histograms in two phase space regions, for multiple physics processes and systematic variations.

```python
utils.set_style()
```

AGC tools 2022 workshop



ServiceX and coffea exploration

- Ongoing studies by Kyungeon Choi on using ServiceX + TRexFitter
- Shown at vCHEP 2021
  - Towards Real-World Applications of ServiceX, an Analysis Data Transformation System
  - arXiv:2107.01789
- Active studies by Kyungeon on ServiceX+coffea (will be shown in ATLAS internal session later today)

→ Much faster to see the result

→ No hassle with local cluster or EOS storage space

vCHEP 2021 (Times are flipped in left plot)

PyHEP ecosystem (Jim's summary)

5 HEP-specific UI applications or packaged algorithms

4 HEP-specific for common problems

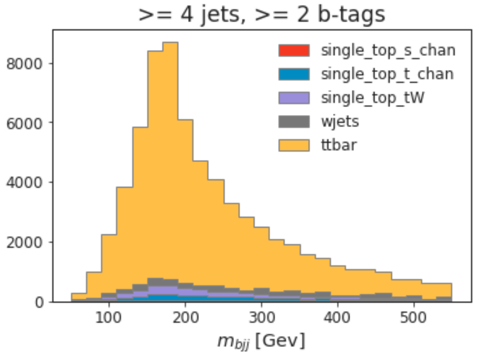
3 HEP-specific, foundational

2 needed to create, but not really HEP-specific

1 non-HEP software we depend on

The AGC analysis setup

- Main AGC analysis task: **ttbar cross-section measurement** in single lepton channel
  - includes simple top quark reconstruction
  - setup chosen as it captures relevant workflow aspects and can easily be extended
    - e.g. conversion into a beyond-the-Standard-Model search
  - analysis task prominently features handling of systematic uncertainties
- Analysis is based on **Run-2 CMS Open Data** (~400 TB of data in MiniAOD format)
  - Open Data is crucial: everyone can participate
  - currently using 4 TB of ntuple inputs (pre-converted, ~1B events before cuts)
- Goal of setup is showing **functionality**, not discovering new physics
  - want to capture *workflow*, but can use made-up tools for evaluating calibrations & systematic uncertainties



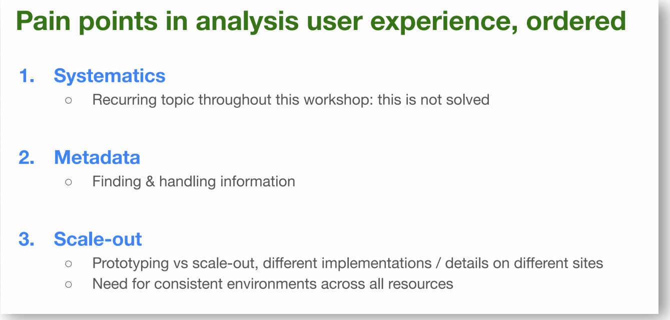
reconstructed top mass

Systematics and other analyzer user experience aspects

- Handling **systematic uncertainties** is a **key challenge** in analysis workflows
  - AGC analysis task includes different types of systematic uncertainties to mirror practical requirements
    - weight-based uncertainties
    - object-based systematic variations affecting kinematics (+ thereby event selection / observables)
    - non-histogram-based uncertainties (e.g. cross-section uncertainties)

- **Metadata** handling
  - capturing various bookkeeping aspects in analysis task

- **Scale-out**: from laptop to analysis facility
  - challenge: write analysis implementation that can run anywhere



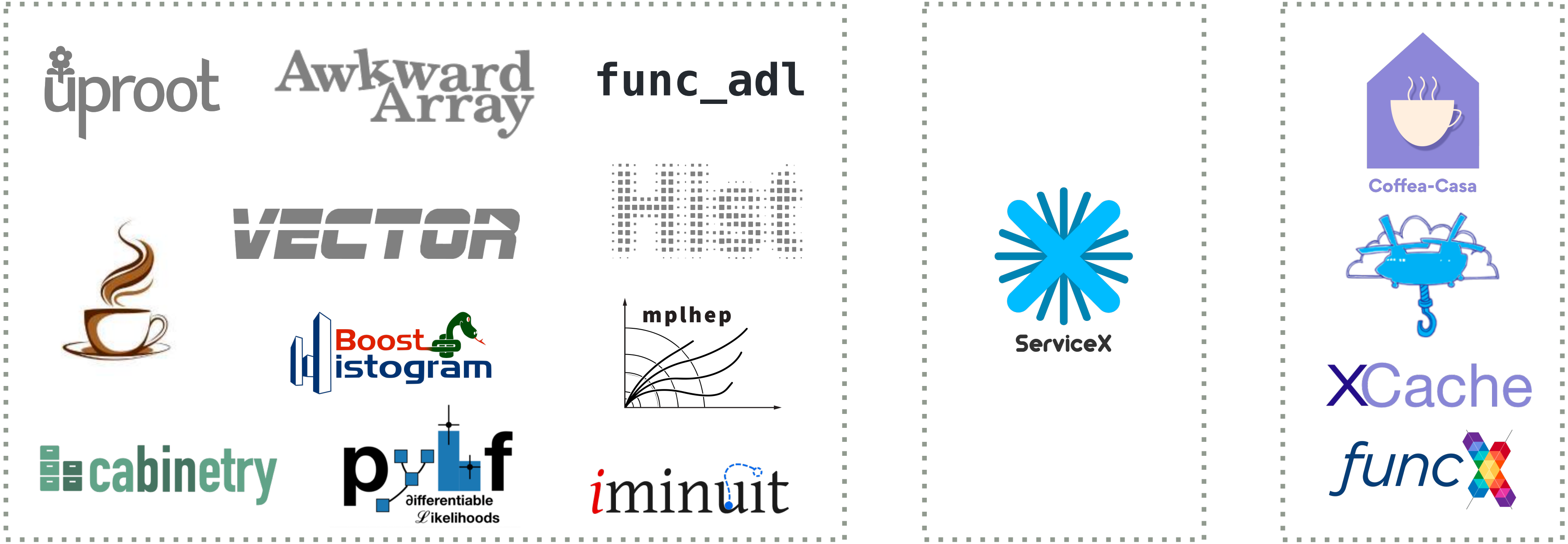
Pain points in analysis user experience, ordered

1. **Systematics**
   - Recurring topic throughout this workshop: this is not solved
2. **Metadata**
   - Finding & handling information
3. **Scale-out**
   - Prototyping vs scale-out, different implementations / details on different sites
   - Need for consistent environments across all resources

Analysis Ecosystem Workwshop II
User experience & Declarative Languages summary

Tools and services in our implementation

- Employing stack of **Python HEP libraries** for analysis tasks
- **ServiceX** used as data delivery service
- Execution on a **coffea-casa analysis facility**



HEP-specific libraries used for data analysis

data delivery services

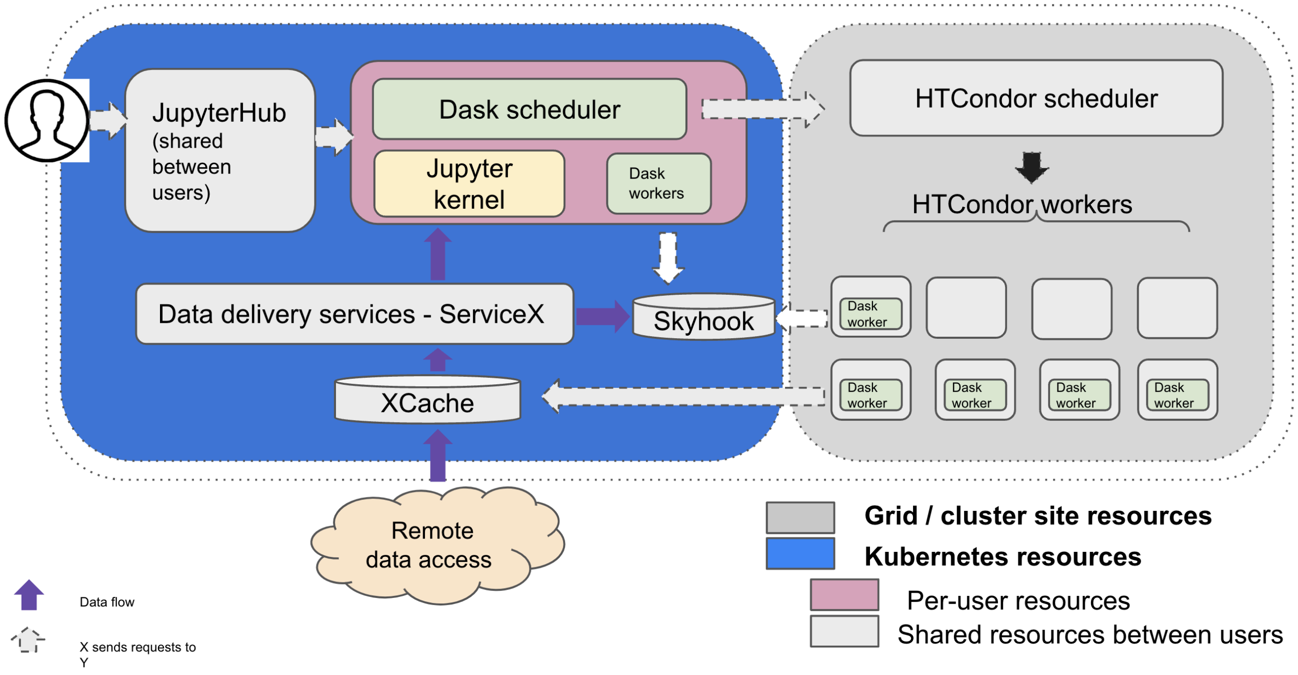
optional services

Analysis Facilities for execution

see O. Shadura's talk for more information!

- **coffea-casa** is a **prototype analysis facility** for the HL-LHC
  - interactive facility for columnar analysis providing analysis tools & scaling to computing resources
  - more information: https://iris-hep.org/projects/coffea-casa.html

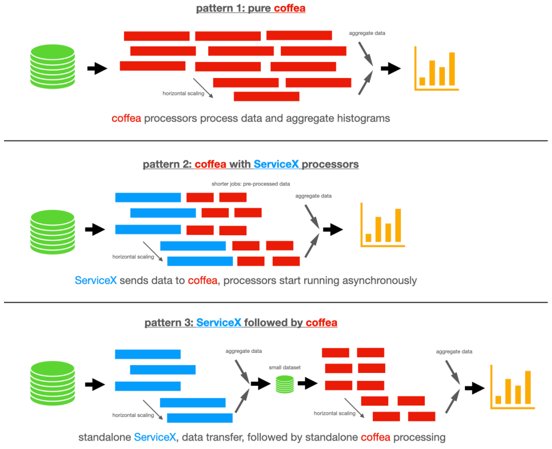
Coffea-Casa



Implementation: ttbar analysis in a notebook

- **From data delivery to statistical inference** in a notebook

multiple supported processing schemes

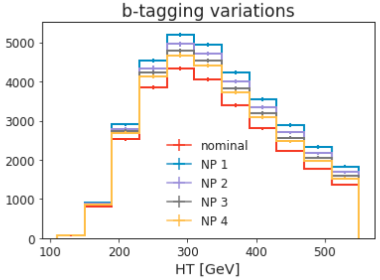


coffea processors process data and aggregate histograms

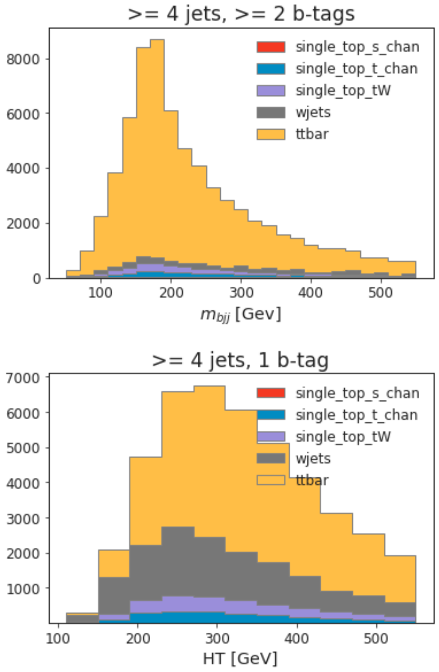
ServiceX sends data to coffea, processors start running asynchronously

standalone ServiceX, data transfer, followed by standalone coffea processing

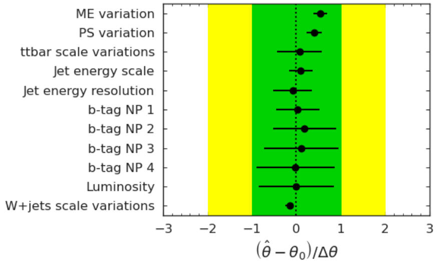
systematic variations



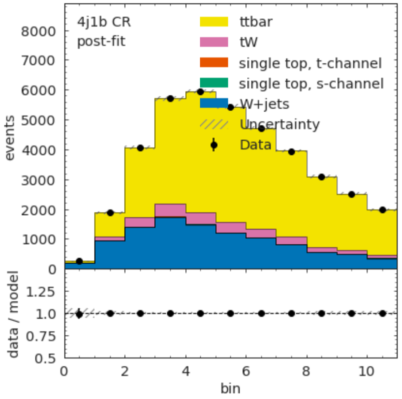
reconstructed observables



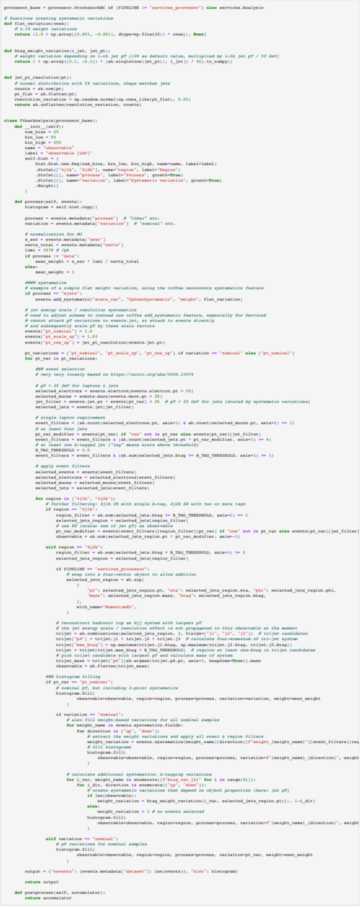
nuisance parameter pulls



post-fit distributions



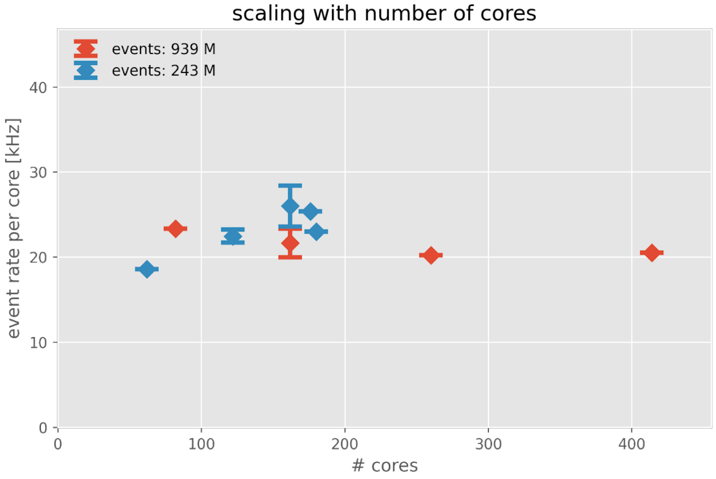
coffea processor



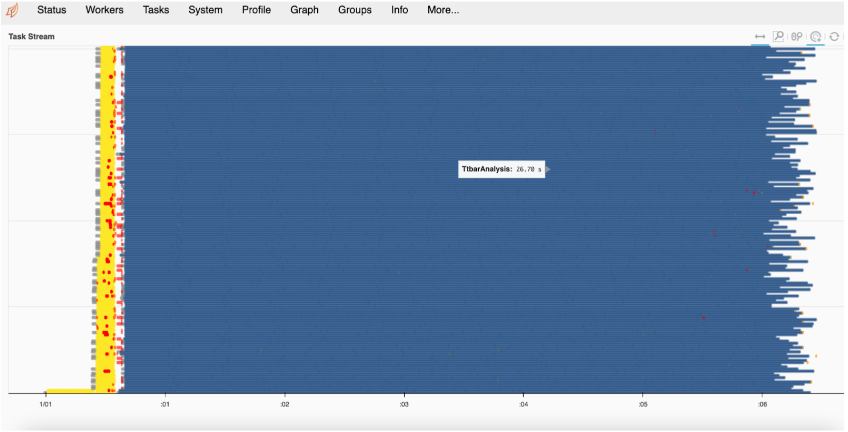
```
processor_base = processor.ProcessorABC if (PIPELINE != "servicex_processor") else servicex.Analysis

# functions creating systematic variations
def flat_variation(ones):
    # 0.1% weight variations
    return (1.0 + np.array([0.001, -0.001], dtype=np.float32)) * ones[:, None]


def btag_weight_variation(i_jet, jet_pt):
    # weight variation depending on i-th jet pT (10% as default value, multiplied by i-th jet pT / 50 GeV)
    return 1 + np.array([0.1, -0.1]) * (ak.singletons(jet_pt[:, i_jet]) / 50).to_numpy()


def jet_pt_resolution(pt):
    # normal distribution with 5% variations, shape matches jets
    counts = ak.num(pt)
    pt_flat = ak.flatten(pt)
    resolution_variation = np.random.normal(np.ones_like(pt_flat), 0.05)
    return ak.unflatten(resolution_variation, counts)


class TtbarAnalysis(processor_base):
    def __init__(self):
        num_bins = 25
        bin_low = 50
        bin_high = 550
        name = "observable"
        label = "observable [GeV]"
        self.hist = (
            hist.Hist.new.Reg(num_bins, bin_low, bin_high, name=name, label=label)
            .StrCat(["4j1b", "4j2b"], name="region", label="Region")
            .StrCat([], name="process", label="Process", growth=True)
            .StrCat([], name="variation", label="Systematic variation", growth=True)
            .Weight()
        )

    def process(self, events):
        histogram = self.hist.copy()

        process = events.metadata["process"]  # "ttbar" etc.
        variation = events.metadata["variation"]  # "nominal" etc.

        # normalization for MC
        x_sec = events.metadata["xsec"]
        nevts_total = events.metadata["nevts"]
        lumi = 3378 # /pb
        if process != "data":
            xsec_weight = x_sec * lumi / nevts_total
        else:
            xsec_weight = 1

        #### systematics
        # example of a simple flat weight variation, using the coffea nanoevents systematics feature
        if process == "wjets":
            events.add_systematic("scale_var", "UpDownSystematic", "weight", flat_variation)

        # jet energy scale / resolution systematics
        # need to adjust schema to instead use coffea add_systematic feature, especially for ServiceX
        # cannot attach pT variations to events.jet, so attach to events directly
        # and subsequently scale pT by these scale factors
        events["pt_nominal"] = 1.0
        events["pt_scale_up"] = 1.03
        events["pt_res_up"] = jet_pt_resolution(events.jet.pt)

        pt_variations = ["pt_nominal", "pt_scale_up", "pt_res_up"] if variation == "nominal" else ["pt_nominal"]
        for pt_var in pt_variations:

            ### event selection
            # very very loosely based on https://arxiv.org/abs/2006.13076

            # pT > 25 GeV for leptons & jets
            selected_electrons = events.electron[events.electron.pt > 25]
            selected_muons = events.muon[events.muon.pt > 25]
            jet_filter = events.jet.pt * events[pt_var] > 25  # pT > 25 GeV for jets (scaled by systematic variations)
            selected_jets = events.jet[jet_filter]

            # single lepton requirement
            event_filters = (ak.count(selected_electrons.pt, axis=1) & ak.count(selected_muons.pt, axis=1) == 1)
            # at least four jets
            pt_var_modifier = events[pt_var] if "res" not in pt_var else events[pt_var][jet_filter]
            event_filters = event_filters & (ak.count(selected_jets.pt * pt_var_modifier, axis=1) >= 4)
            # at least one b-tagged jet ("tag" means score above threshold)
            B_TAG_THRESHOLD = 0.5
            event_filters = event_filters & (ak.sum(selected_jets.btag >= B_TAG_THRESHOLD, axis=1) >= 1)

            # apply event filters
            selected_events = events[event_filters]
            selected_electrons = selected_electrons[event_filters]
            selected_muons = selected_muons[event_filters]
            selected_jets = selected_jets[event_filters]

            for region in ["4j1b", "4j2b"]:
                # further filtering: 4j1b CR with single b-tag, 4j2b SR with two or more tags
                if region == "4j1b":
                    region_filter = ak.sum(selected_jets.btag >= B_TAG_THRESHOLD, axis=1) == 1
                    selected_jets_region = selected_jets[region_filter]
                    # use HT (scalar sum of jet pT) as observable
                    pt_var_modifier = events[event_filters][region_filter][pt_var] if "res" not in pt_var else events[pt_var][jet_filter]
                    observable = ak.sum(selected_jets_region.pt * pt_var_modifier, axis=-1)

                elif region == "4j2b":
                    region_filter = ak.sum(selected_jets.btag > B_TAG_THRESHOLD, axis=1) >= 2
                    selected_jets_region = selected_jets[region_filter]

                    if PIPELINE == "servicex_processor":
                        # wrap into a four-vector object to allow addition
                        selected_jets_region = ak.zip(
                            {
                                "pt": selected_jets_region.pt, "eta": selected_jets_region.eta, "phi": selected_jets_region.phi,
                                "mass": selected_jets_region.mass, "btag": selected_jets_region.btag,
                            },
                            with_name="Momentum4D",
                        )

                    # reconstruct hadronic top as bjj system with largest pT
                    # the jet energy scale / resolution effect is not propagated to this observable at the moment
                    trijet = ak.combinations(selected_jets_region, 3, fields=["j1", "j2", "j3"])  # trijet candidates
                    trijet["p4"] = trijet.j1 + trijet.j2 + trijet.j3  # calculate four-momentum of tri-jet system
                    trijet["max_btag"] = np.maximum(trijet.j1.btag, np.maximum(trijet.j2.btag, trijet.j3.btag))
                    trijet = trijet[trijet.max_btag > B_TAG_THRESHOLD]  # require at least one-btag in trijet candidates
                    # pick trijet candidate with largest pT and calculate mass of system
                    trijet_mass = trijet["p4"][ak.argmax(trijet.p4.pt, axis=1, keepdims=True)].mass
                    observable = ak.flatten(trijet_mass)

                ### histogram filling
                if pt_var == "pt_nominal":
                    # nominal pT, but including 2-point systematics
                    histogram.fill(
                            observable=observable, region=region, process=process, variation=variation, weight=xsec_weight
                        )

                    if variation == "nominal":
                        # also fill weight-based variations for all nominal samples
                        for weight_name in events.systematics.fields:
                            for direction in ["up", "down"]:
                                # extract the weight variations and apply all event & region filters
                                weight_variation = events.systematics[weight_name][direction][f"weight_{weight_name}"][event_filters][reg
                                # fill histograms
                                histogram.fill(
                                    observable=observable, region=region, process=process, variation=f"{weight_name}_{direction}", weight
                                )

                        # calculate additional systematics: b-tagging variations
                        for i_var, weight_name in enumerate([f"btag_var_{i}" for i in range(4)]):
                            for i_dir, direction in enumerate(["up", "down"]):
                                # create systematic variations that depend on object properties (here: jet pT)
                                if len(observable):
                                    weight_variation = btag_weight_variation(i_var, selected_jets_region.pt)[:, 1-i_dir]
                                else:
                                    weight_variation = 1 # no events selected
                                histogram.fill(
                                    observable=observable, region=region, process=process, variation=f"{weight_name}_{direction}", weight
                                )

                elif variation == "nominal":
                    # pT variations for nominal samples
                    histogram.fill(
                            observable=observable, region=region, process=process, variation=pt_var, weight=xsec_weight
                        )

        output = {"nevents": {events.metadata["dataset"]: len(events)}, "hist": histogram}

        return output

    def postprocess(self, accumulator):
        return accumulator
```

Benchmarking results

- **Benchmarking AGC implementation performance** at the University of Nebraska–Lincoln CMS Tier-2
  - tested various configurations of hardware, data pipeline and analysis task
  - for more information, see this ACAT 2022 contribution



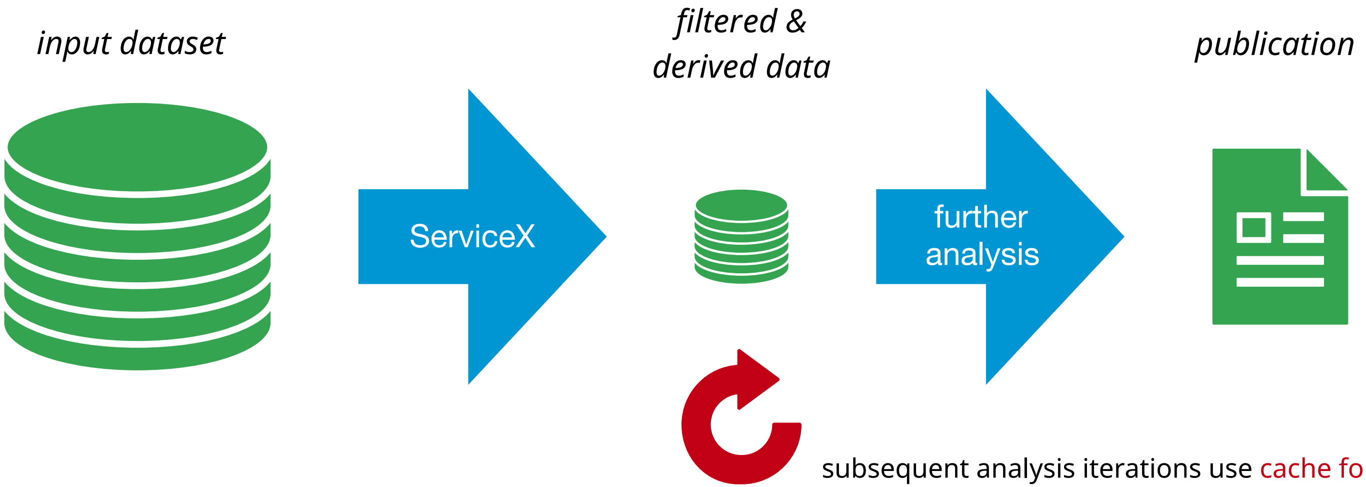
good scaling to hundreds of cores



efficient resource usage via Dask

On-demand columnar data delivery: ServiceX

- ServiceX is a **data extraction and delivery** service
  - users provide list of datasets to process + instructions for how to extract data (e.g. declarative)
  - ServiceX can be co-located with input datasets for fast execution
  - columnar data is returned and cached
    - subsequent executions can use the cache!

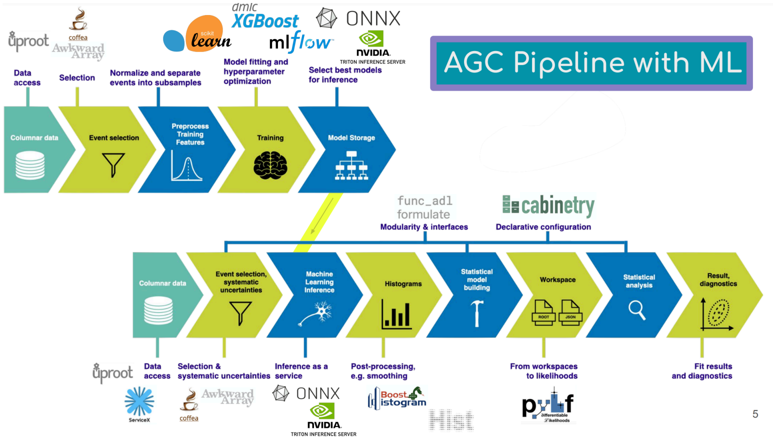


subsequent analysis iterations use cache for significant speedup

AGC "version 2" and future work

- Development of a **"version 2" of the AGC analysis task** is ongoing
  - expanded task: more complexity and data to process
  - inclusion of machine learning aspects (training & inference)

E. Kauffman at AGC demo day #2



AGC Pipeline with ML

uproot
coffea
Awkward Array
scikit learn
dmlc XGBoost
mlflow
ONNX
NVIDIA
TRITON INFERENCE SERVER
Data access
Selection
Normalize and separate events into subsamples
Model fitting and hyperparameter optimization
Select best models for inference
Columnar data
Event selection
Preprocess Training Features
Training
Model Storage
func_adl
formulate
Modularity & interfaces
cabinetry
Declarative configuration
Columnar data
Event selection, systematic uncertainties
Machine Learning Inference
Histograms
Statistical model building
Workspace
ROOT
JSON
Statistical analysis
Result, diagnostics
uproot
Data access
Selection & systematic uncertainties
Inference as a service
Post-processing, e.g. smoothing
From workspaces to likelihoods
Fit results and diagnostics
ServiceX
coffea
Awkward Array
ONNX
NVIDIA
TRITON INFERENCE SERVER
Boost histogram
Hist
pyhf
differentiable likelihoods
5

- **Develop & compare** different **implementations**
  - e.g. implementation using `ROOT RDataFrame`

- **Benchmarking**
  - investigate performance, identify potential additional bottlenecks & implement solutions

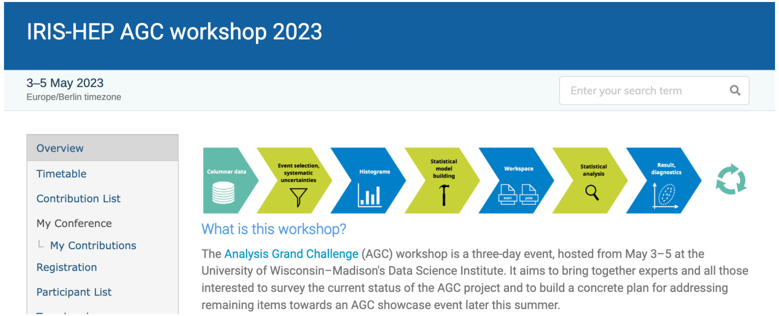
- Longer term plan: **differentiable analysis pipeline**
  - investigate end-to-end analysis optimization, evaluate usefulness vs cost of gradient information

AGC events

- Organizing **yearly workshops**
  - Mix of tutorials, demonstrations, discussions & planning
  - Next workshop: IRIS-HEP AGC workshop on May 3–5

- Recently started bi-monthly **“demo day”** meetings
  - informal, short demos on latest developments
  - broad mix of topics to bring together diverse audience
  - examples: #1, #2 (including recordings)
  - this format works well!

Upcoming AGC workshop in May



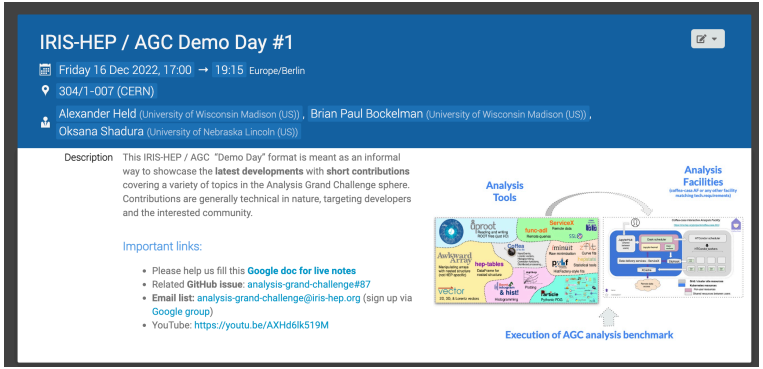
IRIS-HEP AGC workshop 2023

3–5 May 2023
Europe/Berlin timezone

What is this workshop?

The Analysis Grand Challenge (AGC) workshop is a three-day event, hosted from May 3–5 at the University of Wisconsin–Madison's Data Science Institute. It aims to bring together experts and all those interested to survey the current status of the AGC project and to build a concrete plan for addressing remaining items towards an AGC showcase event later this summer.

First AGC "demo day"



IRIS-HEP / AGC Demo Day #1

Friday 16 Dec 2022, 17:00 → 19:15 Europe/Berlin

304/1-007 (CERN)

Alexander Held (University of Wisconsin Madison (US)), Brian Paul Bockelman (University of Wisconsin Madison (US)), Oksana Shadura (University of Nebraska Lincoln (US))

Description: This IRIS-HEP / AGC “Demo Day” format is meant as an informal way to showcase the **latest developments** with **short contributions** covering a variety of topics in the Analysis Grand Challenge sphere. Contributions are generally technical in nature, targeting developers and the interested community.

Important links:

- Please help us fill this **Google doc for live notes**
- Related **GitHub issue**: analysis-grand-challenge#87
- **Email list:** analysis-grand-challenge@iris-hep.org (sign up via Google group)
- YouTube: https://youtu.be/AXHd6lk519M

Summary

- The **Analysis Grand Challenge** is an **integration exercise** to study **HL-LHC analysis workflows**

- Developed **ttbar analysis task & implementation** based on **CMS Open Data**
  - all data & our implementation are publicly available

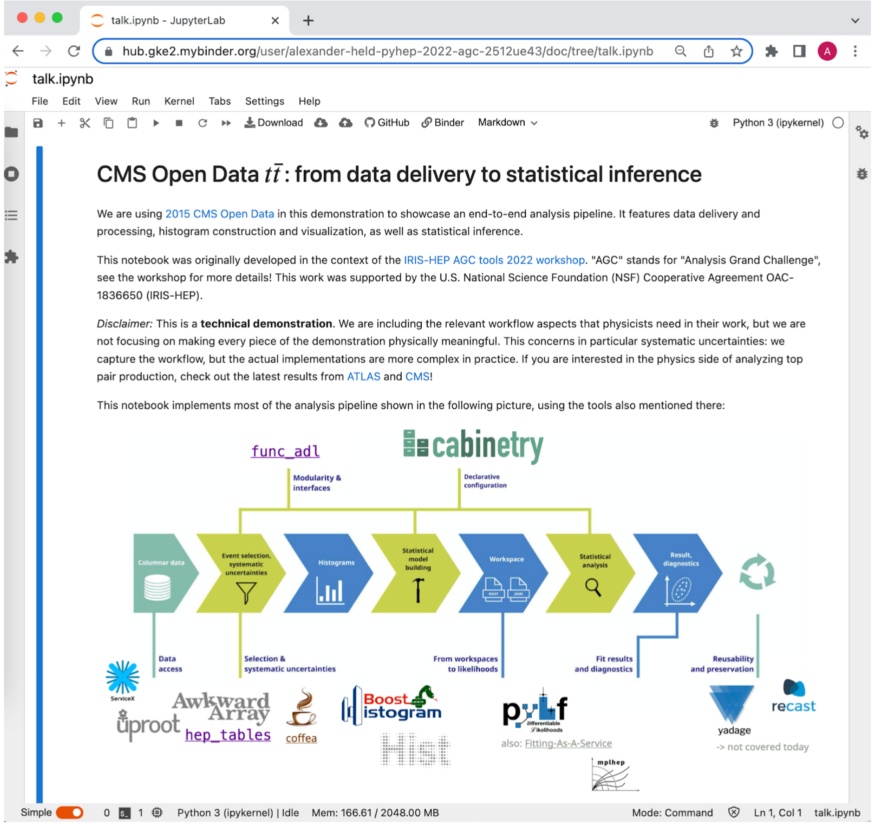
- We hope that the **Analysis Grand Challenge** can be **useful to the broader community**
  - test analysis tools, compare different workflows, test analysis facilities, …

- **Upcoming workshop:** https://indico.cern.ch/e/agc-workshop-2023

- Stay in touch via our **mailing list**
  - analysis-grand-challenge@iris-hep.org (sign up at this Google group)

Give it a try!

- You can **run our AGC** analysis pipeline on **Binder**
  - **Try it out today!**
- All code also available on GitHub
- See also this PyHEP 2022 contribution
  - includes recording of walkthrough



CMS Open Data $t\bar{t}$: from data delivery to statistical inference

We are using 2015 CMS Open Data in this demonstration to showcase an end-to-end analysis pipeline. It features data delivery and processing, histogram construction and visualization, as well as statistical inference.

This notebook was originally developed in the context of the IRIS-HEP AGC tools 2022 workshop. "AGC" stands for "Analysis Grand Challenge", see the workshop for more details! This work was supported by the U.S. National Science Foundation (NSF) Cooperative Agreement OAC-1836650 (IRIS-HEP).

Disclaimer: This is a **technical demonstration**. We are including the relevant workflow aspects that physicists need in their work, but we are not focusing on making every piece of the demonstration physically meaningful. This concerns in particular systematic uncertainties: we capture the workflow, but the actual implementations are more complex in practice. If you are interested in the physics side of analyzing top pair production, check out the latest results from ATLAS and CMS!

This notebook implements most of the analysis pipeline shown in the following picture, using the tools also mentioned there:

Thank you!

- The **AGC is made possible** thanks to the **help of a large number of people** working on many different projects.

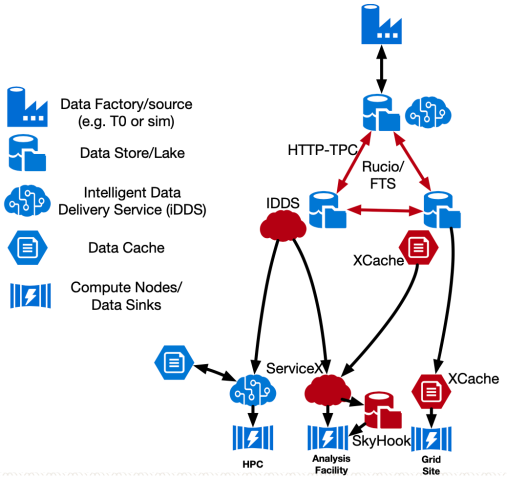
- **Thank you** in particular to the teams behind:
  - coffea-casa
  - Scikit-HEP, coffea, IRIS-HEP Analysis Systems
  - ServiceX, IRIS-HEP DOMA
  - IRIS-HEP SSL
  - CMS Open Data

Backup

Integration: connecting IRIS-HEP focus areas

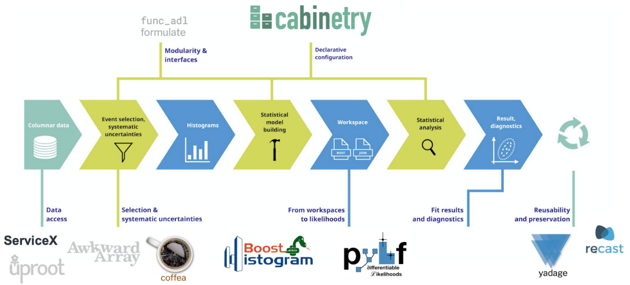
Data Organization, Management and Access (DOMA):

data delivery



Analysis Systems (AS):

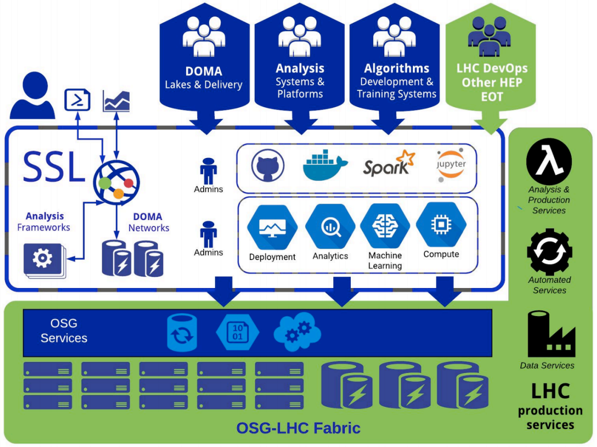
tools



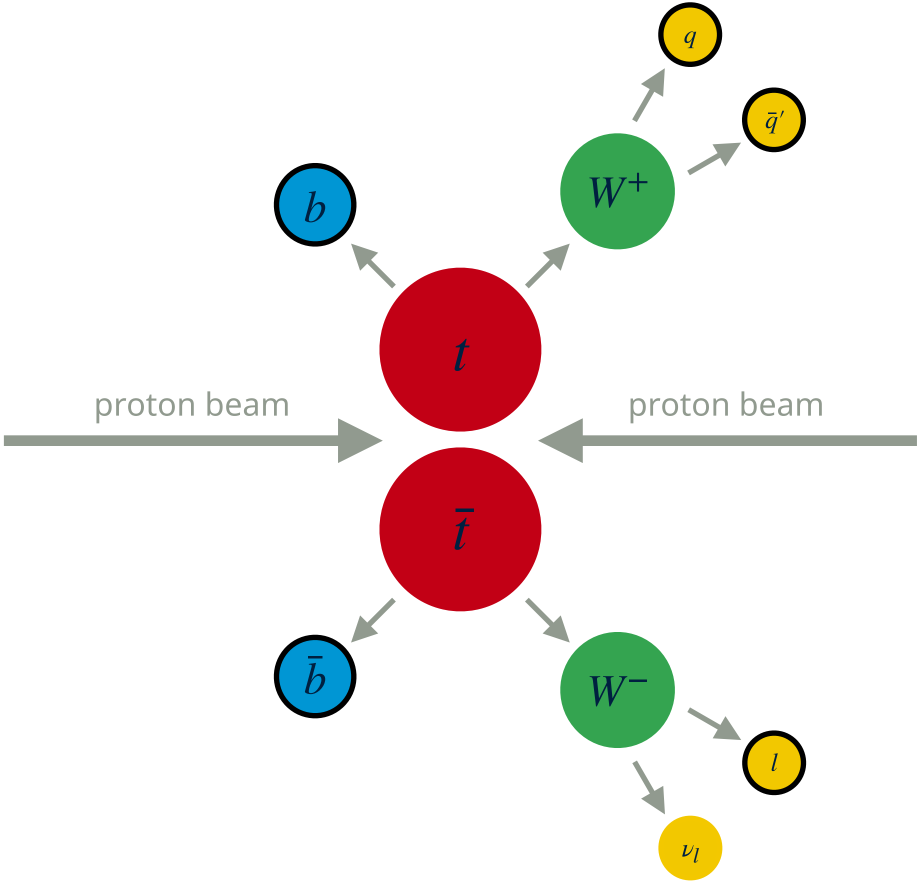
func_adl
formulate
cabinetry
Modularity & interfaces
Declarative configuration
Columnar data
Event selection, systematic uncertainties
Histograms
Statistical model building
Workspace
ROOT
JSON
Statistical analysis
Result, diagnostics
Data access
Selection & systematic uncertainties
From workspaces to likelihoods
Fit results and diagnostics
Reusability and preservation
ServiceX
uproot
Awkward Array
coffea
Boost histogram
pyhf
differentiable likelihoods
yadage
recast

Scalable Systems Laboratory (SSL):

deployment techniques and resources



Top quark pair production


q
$\bar{q}'$
b
W^+
t
proton beam
proton beam
$\bar{t}$
$\bar{b}$
W^-
l
ν_l