IRIS-HEP analysis pipeline

Alexander Held[1]

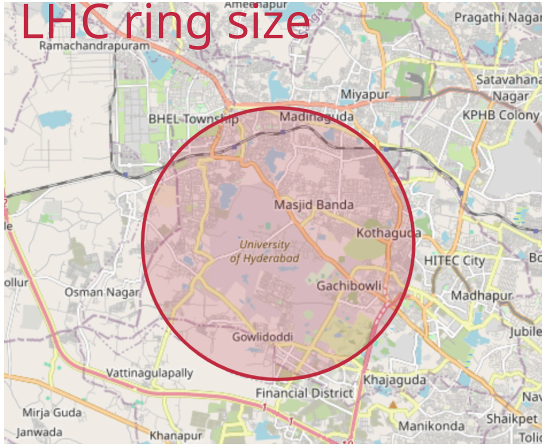
[1] University of Wisconsin-Madison

HSF India Hyderabad
https://indico.cern.ch/event/1394564/
Jan 17, 2025

This work was supported by the U.S. National Science Foundation (NSF) under Cooperative Agreements OAC-1836650 and PHY-2323298.

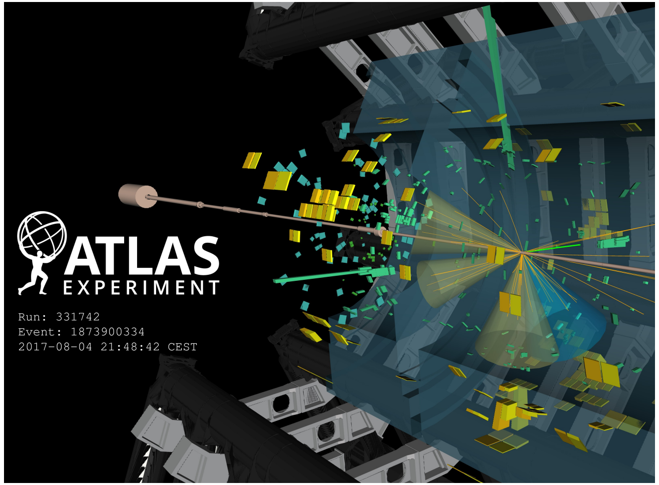
The big picture: collision to publication

1) collide protons



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

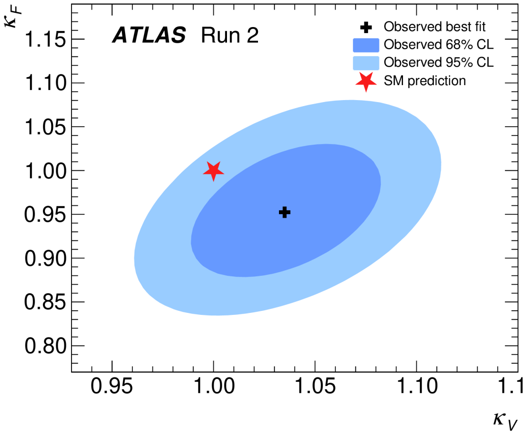
2) observe remnants



Phys. Lett. B 784 (2018) 173

O(100 M) files with
O(100 B) events
(data + simulation)

3) infer nature

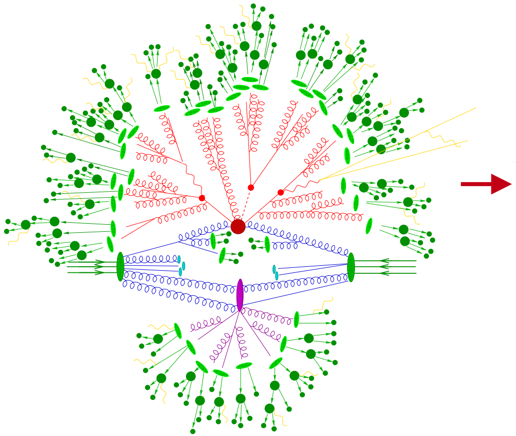


Nature 607, 52–59 (2022)

O(1000) sources of uncertainty

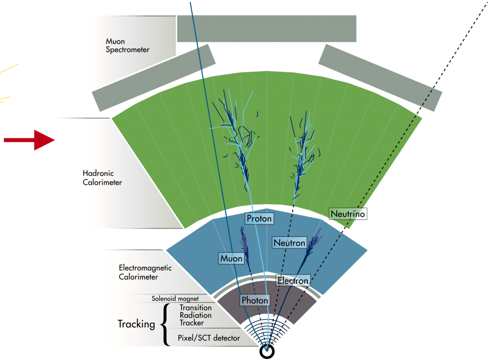
End-user physics analysis

event generation



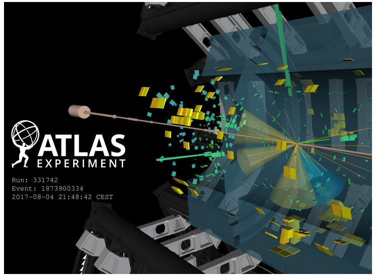
JHEP 0902 (2009) 007

detector interaction



CERN-EX-1301009

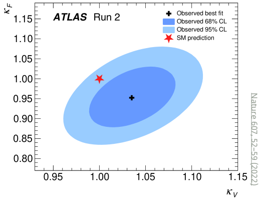
object reconstruction



Phys. Lett. B 784 (2018) 173

“analysis”

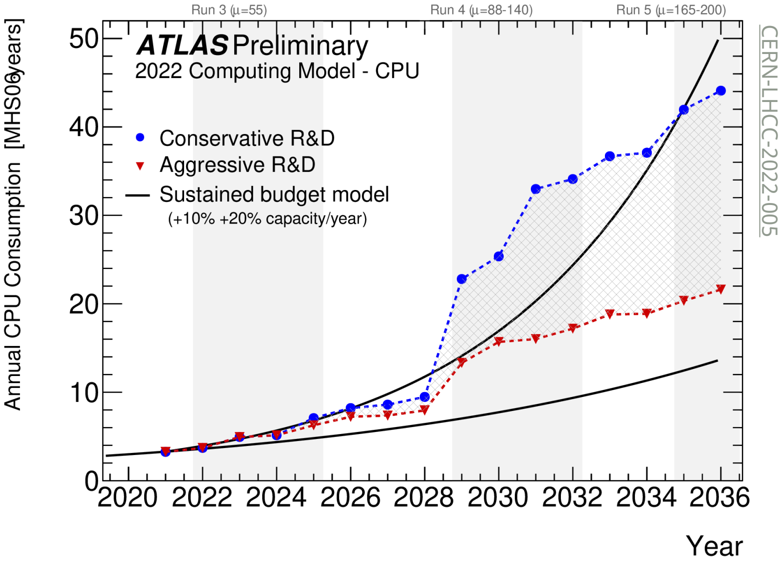
centrally provided datasets

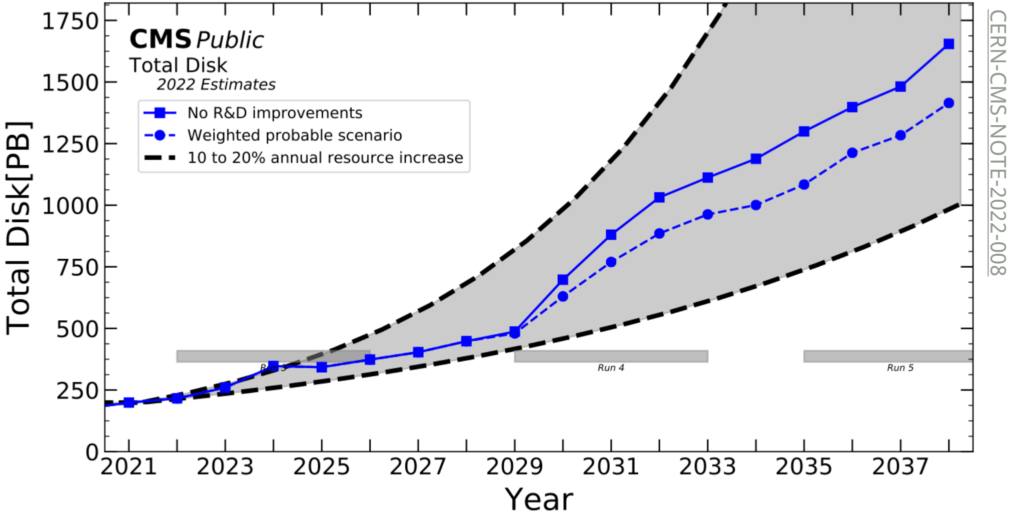


Nature 607, 52–59 (2022)

- “Analysis” in practice: the whole pipeline **turning centrally provided datasets into results for a paper**
  - iterative process, optimize, debug, validate: low latency means faster time-to-insight

Computing challenges



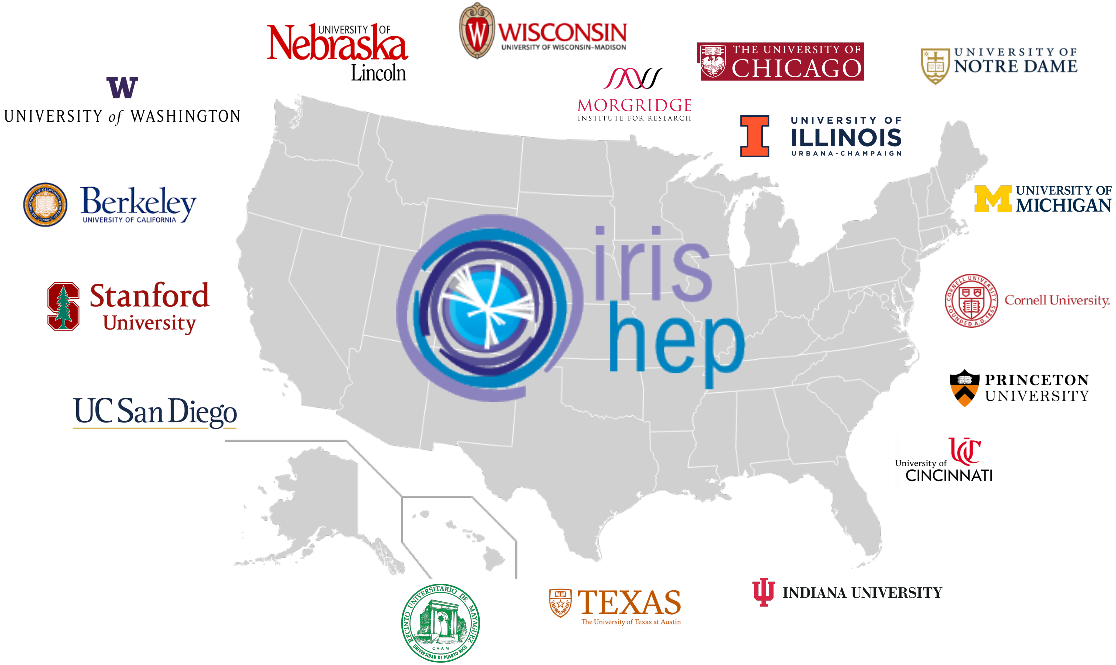


IRIS-HEP and a HL-LHC vision

The IRIS-HEP software institute

- **IRIS-HEP**: *"Institute for Research and Innovation in Software for High Energy Physics"*
  - a software institute funded by the US National Science Foundation, running 2018–2028
  - working in close collaboration with LHC experiments and facilities

U.S. National Science Foundation



University of Nebraska Lincoln
Wisconsin University of Wisconsin-Madison
The University of Chicago
University of Notre Dame
University of Washington
Morgridge Institute for Research
University of Illinois Urbana-Champaign
Berkeley University of California
University of Michigan
iris hep
Stanford University
Cornell University
Princeton University
UC San Diego
University of Cincinnati
Recinto Universitario de Mayagüez Universidad de Puerto Rico
Texas The University of Texas at Austin
Indiana University

[2024 IRIS-HEP retreat]

R&D for the HL-LHC

- IRIS-HEP is working on **computing and software R&D for the HL-LHC**
  - a software upgrade accompanying detector hardware upgrades
  - focusing on a subset of activity areas today, connected through “challenge” format

DOMA

data organisation and management

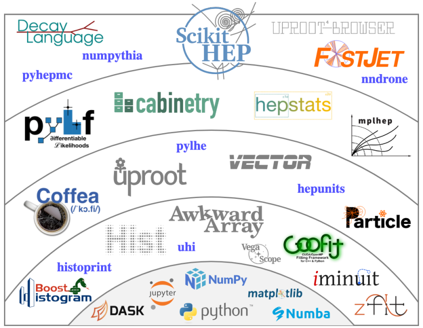
ServiceX XCache

Coffea-Casa

WLCG Data Challenge

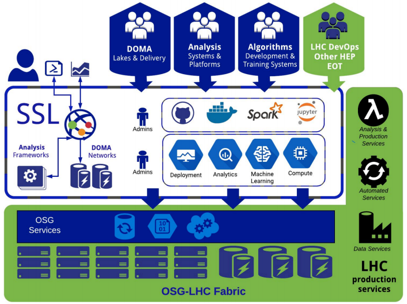
AS

analysis systems and tools



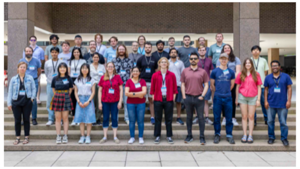
SSL and OSG-LHC

deployment techniques and resources



SSC

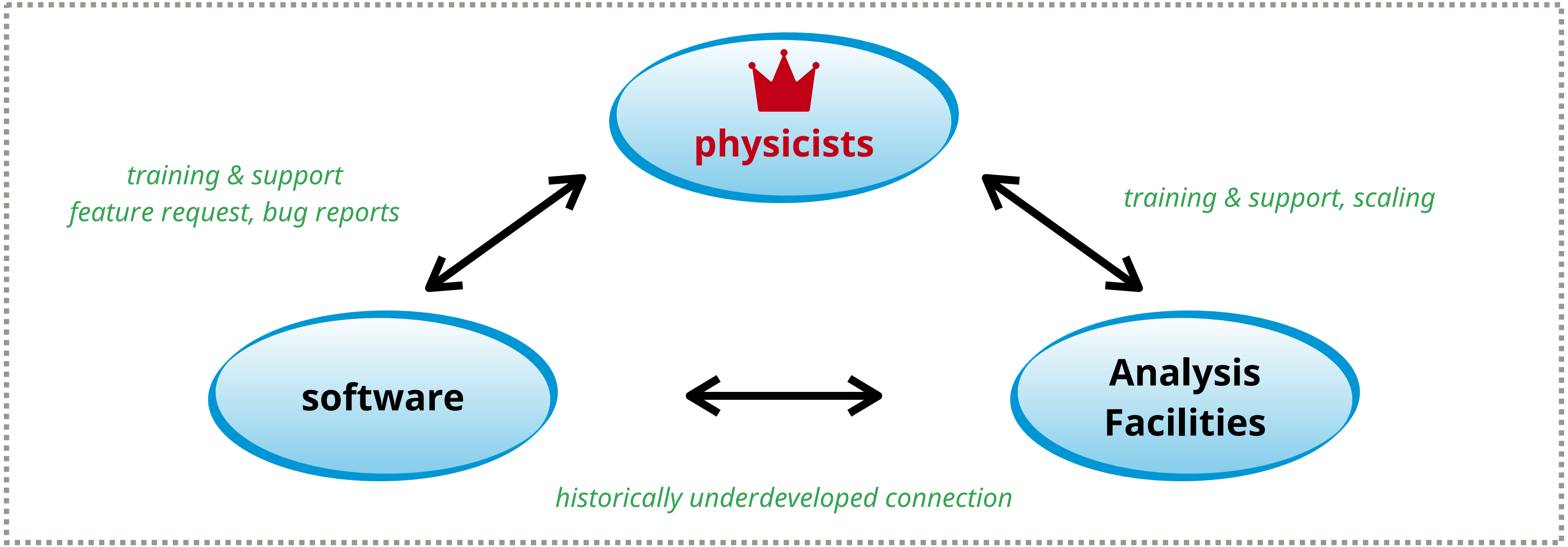
training



Empowering physicists, today and tomorrow

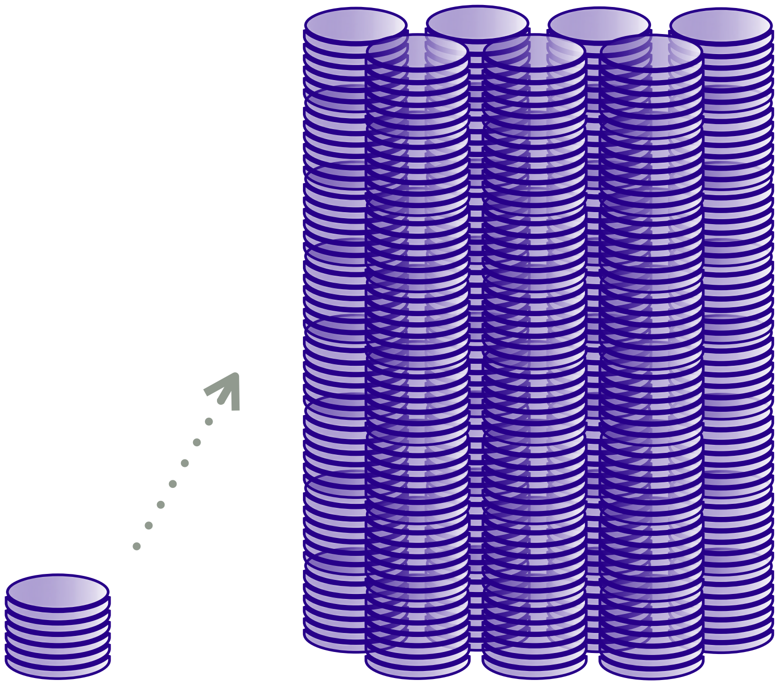
- This work is driven by the desire to **minimize time-to-insight** and **maximize the HL-LHC physics reach**
  - let **physicists** spend more time doing physics, less time debugging, bookkeeping, waiting, …
  - tighten feedback & support cycles by connecting communities together
- **Physics is the end goal**: strive to find ways to overcome computing challenges

IRIS-HEP is active in all these areas


physicists
training & support
feature request, bug reports
training & support, scaling
software
Analysis Facilities
historically underdeveloped connection

Our end-user analysis vision

- **Analyze O(1000) TB of data within a few hours**
- Interactive analysis turnaround: “coffee break” timescale
- Fully integrated Analysis Facilities (AFs)
- UX to empower big & small teams
- Easy access to state-of-the-art ML + techniques
- Reproducibility, preservation, reuse



today:

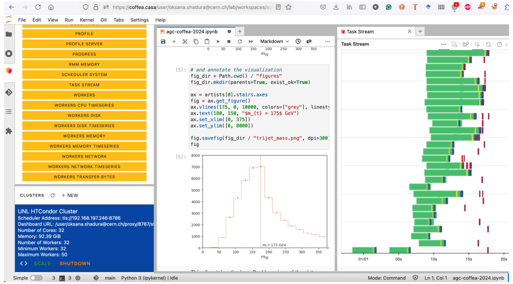
create $\mathcal{O}(1-10)$ TB ntuples
on the grid
in $\mathcal{O}$(days – weeks),
analyze on Tier-3 in $\mathcal{O}(h$ – days)

HL-LHC:

analyze $\mathcal{O}(1000)$ TB of data
straight out of central
PHYSLITE / NanoAOD files in $\mathcal{O}(h)$

Our end-user analysis vision

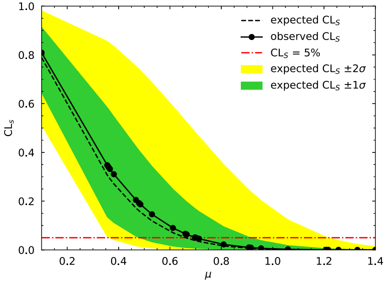
- Analyze O(1000) TB of data within a few hours
- **Interactive analysis turnaround: “coffee break” timescale**
- Fully integrated Analysis Facilities (AFs)
- UX to empower big & small teams
- Easy access to state-of-the-art ML + techniques
- Reproducibility, preservation, reuse



UNL HTCondor Cluster
Scheduler Address: tls://192.168.197.246:8786
Dashboard URL: /user/oksana.shadura@cern.ch/proxy/8787/s
Number of Cores: 32
Memory: 92.39 GiB
Number of Workers: 32
Minimum Workers: 32
Maximum Workers: 50
SCALE SHUTDOWN

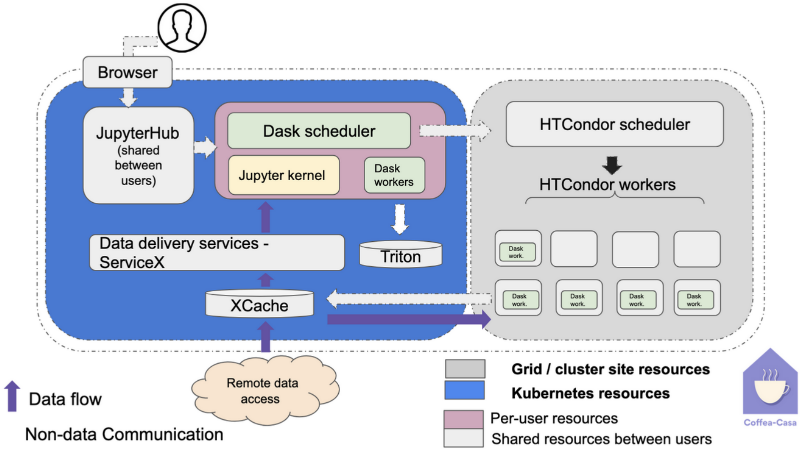
```
[5]: # and annotate the visualization
     fig_dir = Path.cwd() / "figures"
     fig_dir.mkdir(parents=True, exist_ok=True)

     ax = artists[0].stairs.axes
     fig = ax.get_figure()
     ax.vlines(175, 0, 10000, colors=["grey"], linesty
     ax.text(180, 150, "$m_{t} = 175$ GeV")
     ax.set_xlim([0, 375])
     ax.set_ylim([0, 8000])

     fig.savefig(fig_dir / "trijet_mass.png", dpi=300
     fig
```

meaningful analysis iterations on timescale of a coffee break, interactive analysis design



Our end-user analysis vision

- Analyze O(1000) TB of data within a few hours
- Interactive analysis turnaround: “coffee break” timescale
- **Fully integrated Analysis Facilities (AFs)**
- UX to empower big & small teams
- Easy access to state-of-the-art ML + techniques
- Reproducibility, preservation, reuse



required services available,
convenient interfaces,
access to powerful resources

Our end-user analysis vision

- Analyze O(1000) TB of data within a few hours
- Interactive analysis turnaround: “coffee break” timescale
- Fully integrated Analysis Facilities (AFs)
- **UX to empower big & small teams**
- Easy access to state-of-the-art ML + techniques
- Reproducibility, preservation, reuse

efficient collaboration within analysis team

software environment handling

data access & output sharing

fast turnaround

seamless laptop-to-AF/grid transition

see also: ***HSF AF White Paper***
https://arxiv.org/abs/2404.02100

Our end-user analysis vision

- Analyze O(1000) TB of data within a few hours

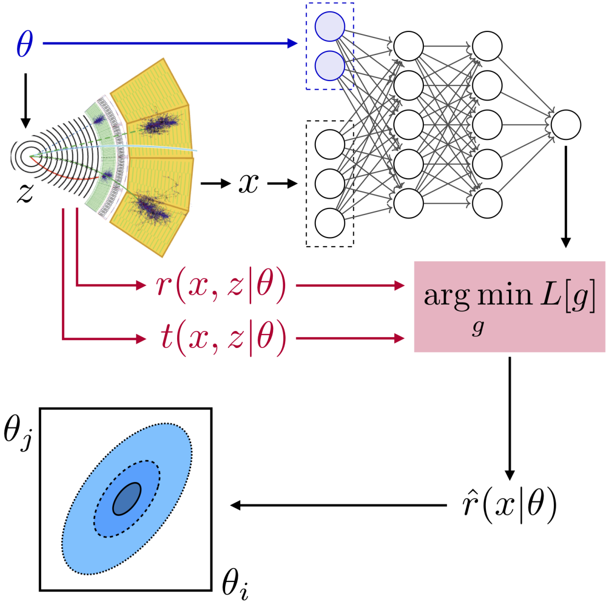
- Interactive analysis turnaround: “coffee break” timescale

- Fully integrated Analysis Facilities (AFs)

- UX to empower big & small teams

- **Easy access to state-of-the-art ML + techniques**

- Reproducibility, preservation, reuse



from MadMiner tutorial

simulation-based inference techniques use different workflows from traditional histogram-based approaches

Our end-user analysis vision

- Analyze O(1000) TB of data within a few hours

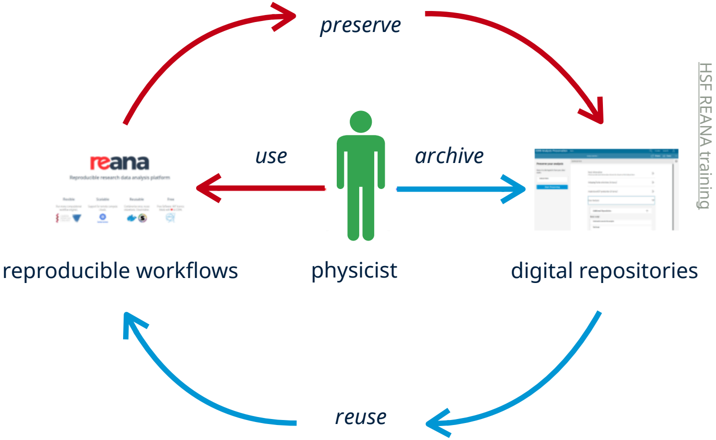
- Interactive analysis turnaround: “coffee break” timescale

- Fully integrated Analysis Facilities (AFs)

- UX to empower big & small teams

- Easy access to state-of-the-art ML + techniques

- **Reproducibility, preservation, reuse**



preserve

use

archive

reproducible workflows

physicist

digital repositories

reuse

HSF REANA training

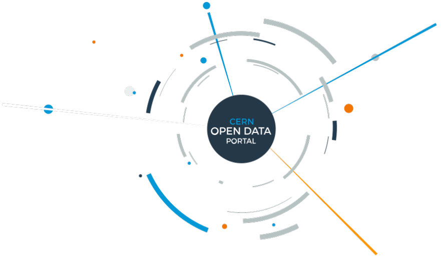
sustainable research
maximizing long-term impact and legacy

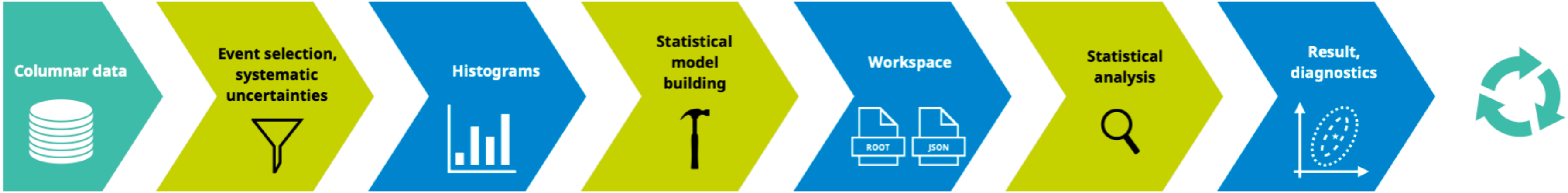
see also:
Nature 533, 452–454 (2016)
arXiv:2203.10057 [hep-ph]

The Analysis Grand Challenge (AGC)

A test case for HL-LHC analysis

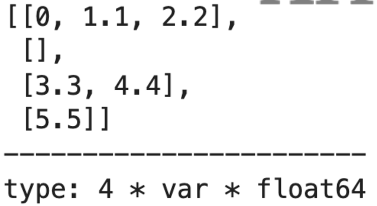
- The **Analysis Grand Challenge (AGC)** defines a **physics analysis task** with **Open Data** to test **HL-LHC workflows**
  - columnar data extraction from large datasets & data processing into histograms
  - statistical model construction and statistical inference, relevant visualizations
  - ML training & inference





Columnar analysis with awkward & coffea

- **awkward**: handles **nested, variable-size data** with numpy-like interface
  - crucial generalization for HEP: different number of objects observed per event



```
[[0, 1.1, 2.2],
 [],
 [3.3, 4.4],
 [5.5]]
------------------------
type: 4 * var * float64
```

- **coffea**: an **interface** to the HEP Python stack which **fills in the gaps** & provides additional **convenient functionality**
  - "schemas":

    ```
    jet_pt
    jet_eta
    jet_phi
    …
    ```

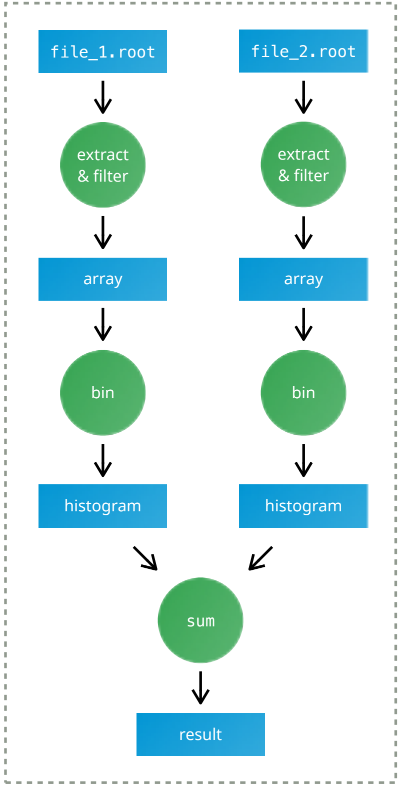
    → `jet` (with properties `jet.pt` etc.)

    - handle physics objects like jets instead of lots of independent arrays
    - many convenience features leading to concise code: `(jets[:, 0] + jets[:, 1]).mass`
  - plus a lot more, e.g. convenient interfaces to ML inference tooling

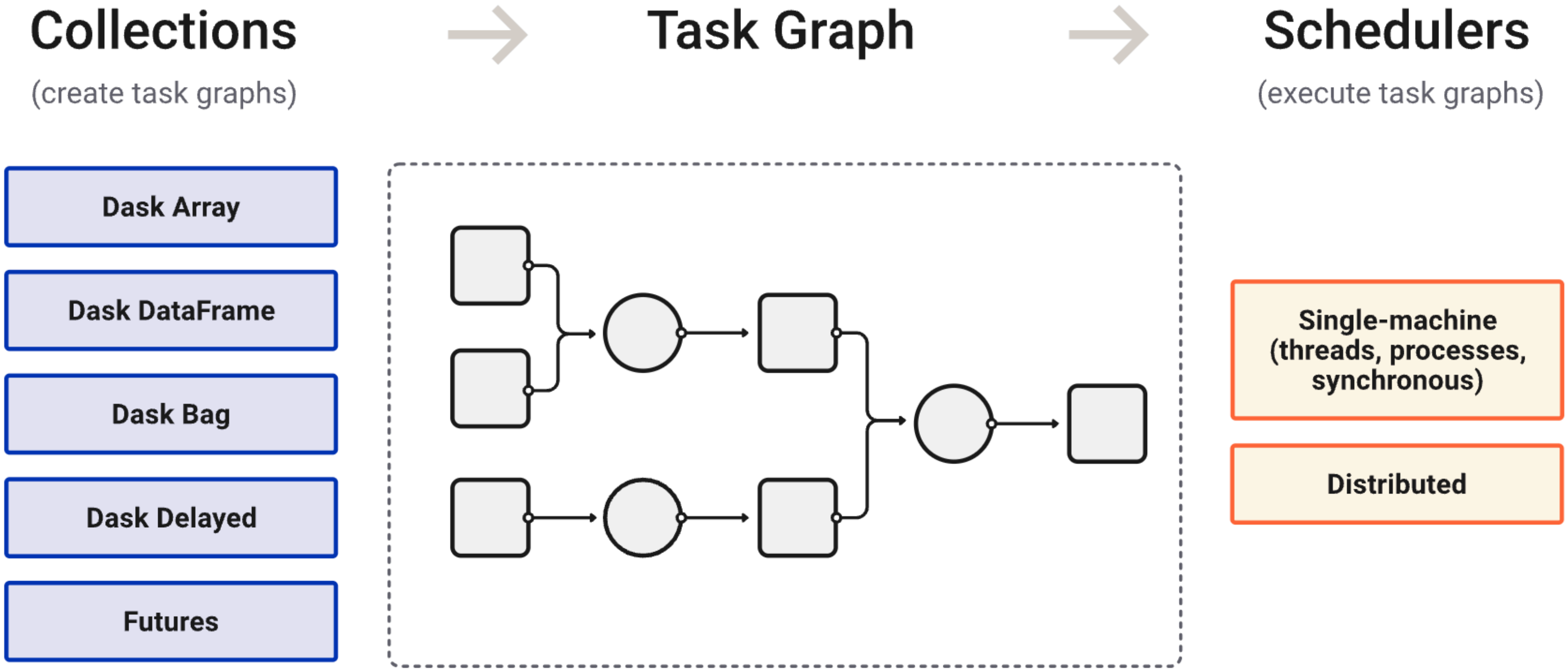
Analysis with task graphs and dask

- We employ **task graphs** to **express & execute** data analysis operations
- This relies on **dask**, a **Python library** providing
  - an interface to describe manipulations of data via task graphs
  - a task scheduler to execute task graphs
- **Deep integration of Dask** and existing **Python HEP toolset** with minimal API changes
  - arrays via **dask-awkward**, histograms, coffea etc.
  - Dask emerging as common feature in Analysis Facilities

example of a task graph



High-level Dask picture



Collections

(create task graphs)

Task Graph

Schedulers

(execute task graphs)

Dask Array

Dask DataFrame

Dask Bag

Dask Delayed

Futures

Single-machine (threads, processes, synchronous)

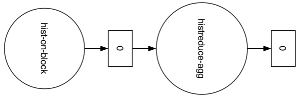
Distributed

High level collections are used to generate task graphs which can be executed by schedulers on a single machine or a cluster.

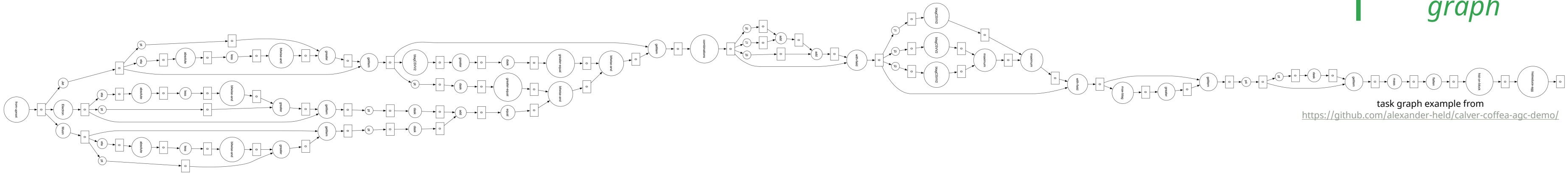
https://docs.dask.org/en/stable/10-minutes-to-dask.html

Analysis as task graphs

- This is a **specific way of thinking about analysis**: *separated intent and compute*
  - take graph, run anywhere (done e.g. with plugins for batch systems)
  - built-in graph optimization is available -> efficient execution (e.g. reduce data transfers)
  - type information immediately available: catch bugs before execution time



optimize graph



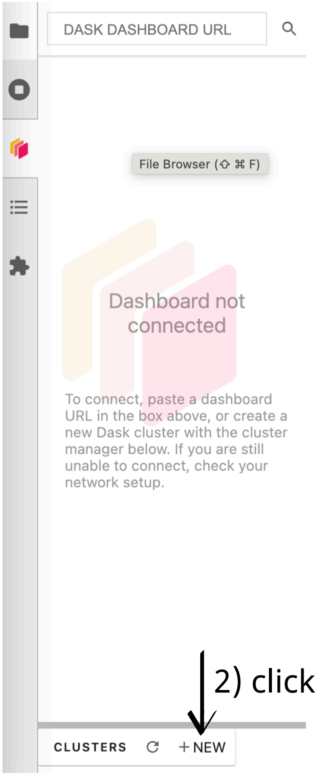
task graph example from
https://github.com/alexander-held/calver-coffea-agc-demo/

- See also: **workflow management systems**
  - higher-level analysis organization: can specify full workflow from central files to paper plots
  - specify exact environments & all implicit assumptions
  - close connections to analysis preservation & reinterpretation

Hands-on with Dask

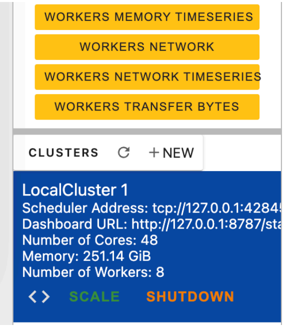
- Launch https://binderhub.ssl-hep.org/v2/gh/research-software-collaborations/courses-hsf-india-january2025/HEAD

1) click



To connect, paste a dashboard URL in the box above, or create a new Dask cluster with the cluster manager below. If you are still unable to connect, check your network setup.

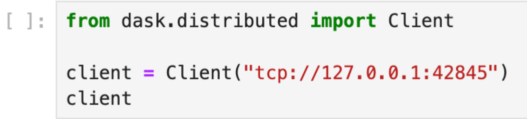
2) click



LocalCluster 1
Scheduler Address: tcp://127.0.0.1:4284
Dashboard URL: http://127.0.0.1:8787/sta
Number of Cores: 48
Memory: 251.14 GiB
Number of Workers: 8

3) click and hold box,
drag into notebook

4) you should see a cell like this



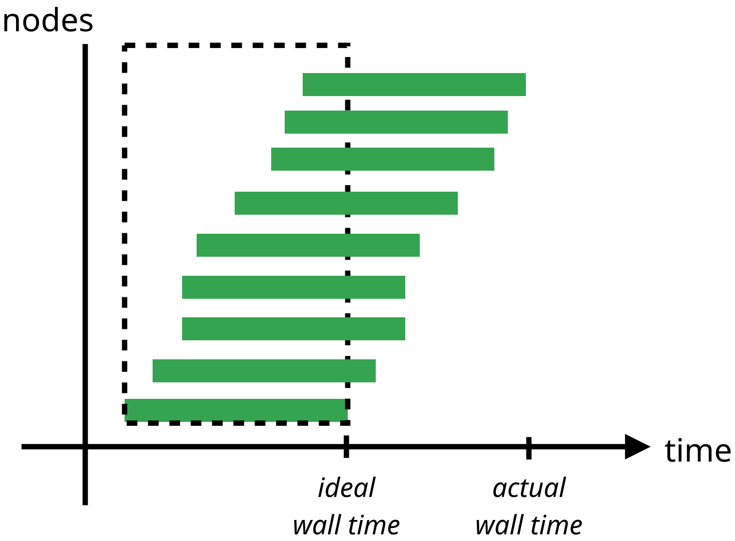
```
[ ]: from dask.distributed import Client

     client = Client("tcp://127.0.0.1:42845")
     client
```

Benefitting from the broader ecosystem

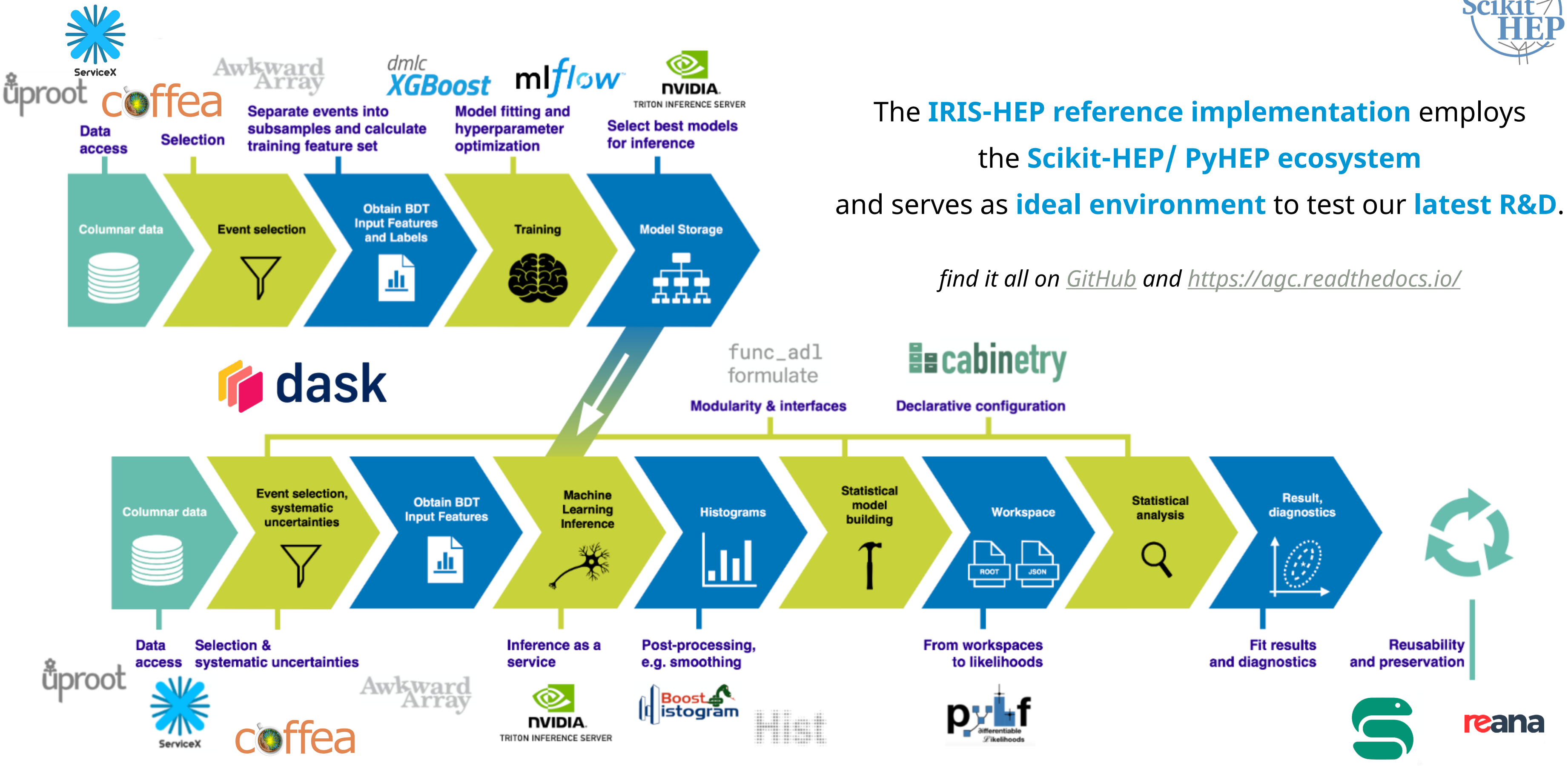
- There is more to low-latency analysis than constructing optimized task graph — **task scheduling can matter a lot!**
  - even with embarrassingly parallel workload, efficient scheduling is crucial for latency

- **Benefit from the broader ecosystem** surrounding Dask: more schedulers are available
  - example: integrating `TaskVine` to schedule coffea-based workload: **~20x wall time improvement** (more details)

TaskVine



The IRIS-HEP AGC reference implementation



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

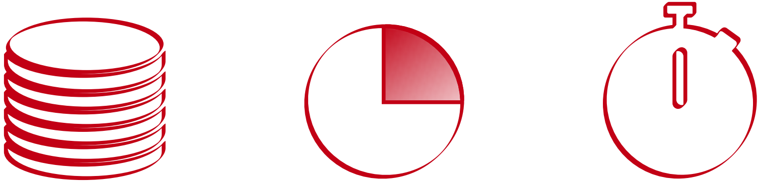
The **IRIS-HEP reference implementation** employs the **Scikit-HEP/ PyHEP ecosystem** and serves as **ideal environment** to test our **latest R&D**.

find it all on GitHub and https://agc.readthedocs.io/

The 200 Gbps Challenge

Defining the 200 Gbps Challenge

200 Gbps →



180 TB dataset　*25% of file content*　*30 minutes*

Reaching these scales poses significant challenges. We set ourselves an ambitious goal.

… and had only 8 weeks to reach it.

CMS NanoAOD example

With **2 kB / event**, this means

- **90 B events,**
- **50 MHz event rate,**

or 1k cores with 50 kHz and 25 MB/s each.

ATLAS PHYSLITE example

With **10 kB / event**, this means

- **18 B events,**
- **10 MHz event rate,**

or 2k cores with 5 kHz and 12.5 MB/s each.

Defining the 200 Gbps Challenge

200 Gbps →

180 TB dataset

25% of file content

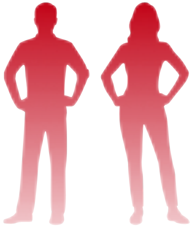
30 minutes

- Targeting **“HL-LHC scale” analysis**, including decompression & data in memory as arrays

- **Two different setups, targeting realism**, all code on GitHub
    - Nebraska: analyze Run-3 NanoAOD CMS data (iris-hep/idap-200gbps)
    - UChicago: analyze Run-2 PHYSLITE ATLAS data (iris-hep/idap-200gbps-atlas)
    - similar tasks broadly, important differences: facilities, event sizes, object types, compression, …

Ingredients for 200 Gbps throughout

planning, structure, schedule



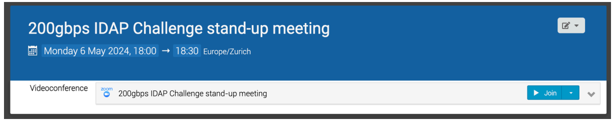
team of experts from IRIS-HEP and beyond,
rallied behind a shared vision

100s of messages per day,
dedicated communication channels

slack

[image by DALL·E 3]

dedicated meetings in multiple formats



200gbps IDAP Challenge stand-up meeting

Monday 6 May 2024, 18:00 → 18:30 Europe/Zurich

Videoconference: 200gbps IDAP Challenge stand-up meeting

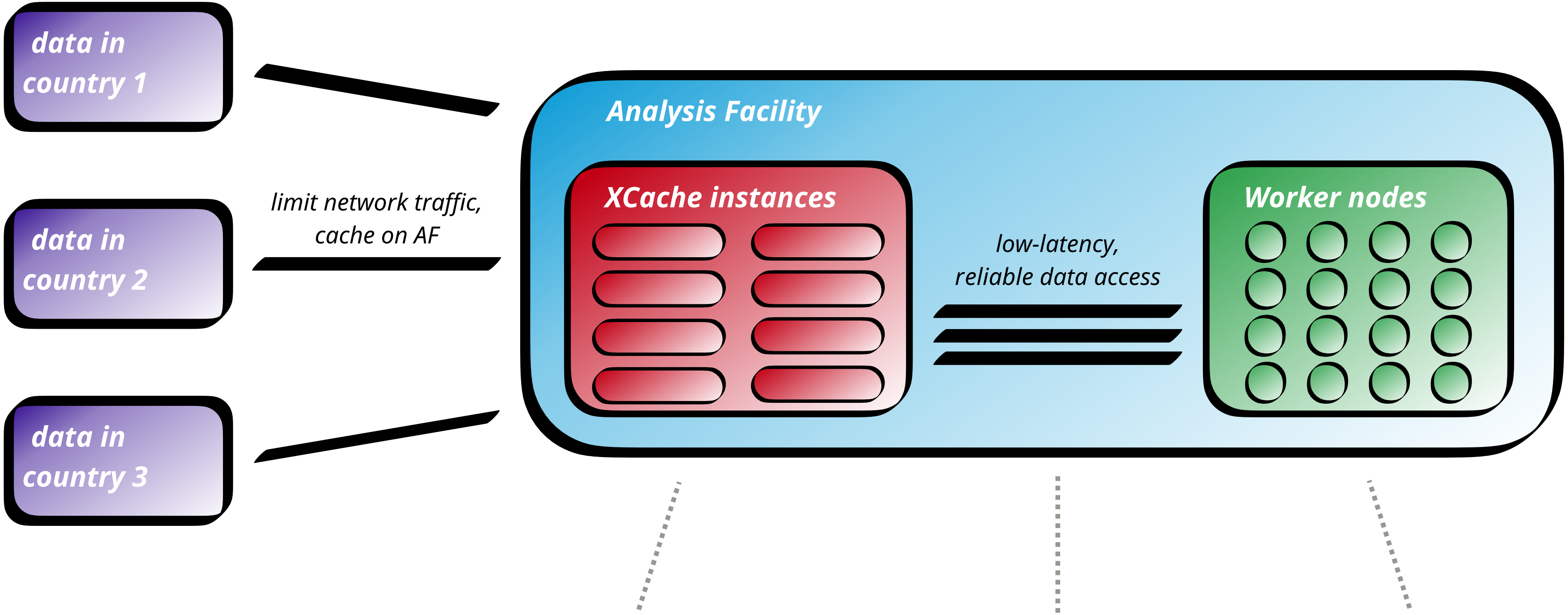
challenging timeline: ***8 weeks***
from first idea to WLCG/HSF workshop

WLCG/HSF Workshop 2024

Demonstrator Analysis 200 Gb/s — *Brian Paul Bockelman*

Hoersaal, DESY — 16:00 - 16:20

Key Analysis Facility elements for 200 Gbps



cache delivering 200+ Gbps

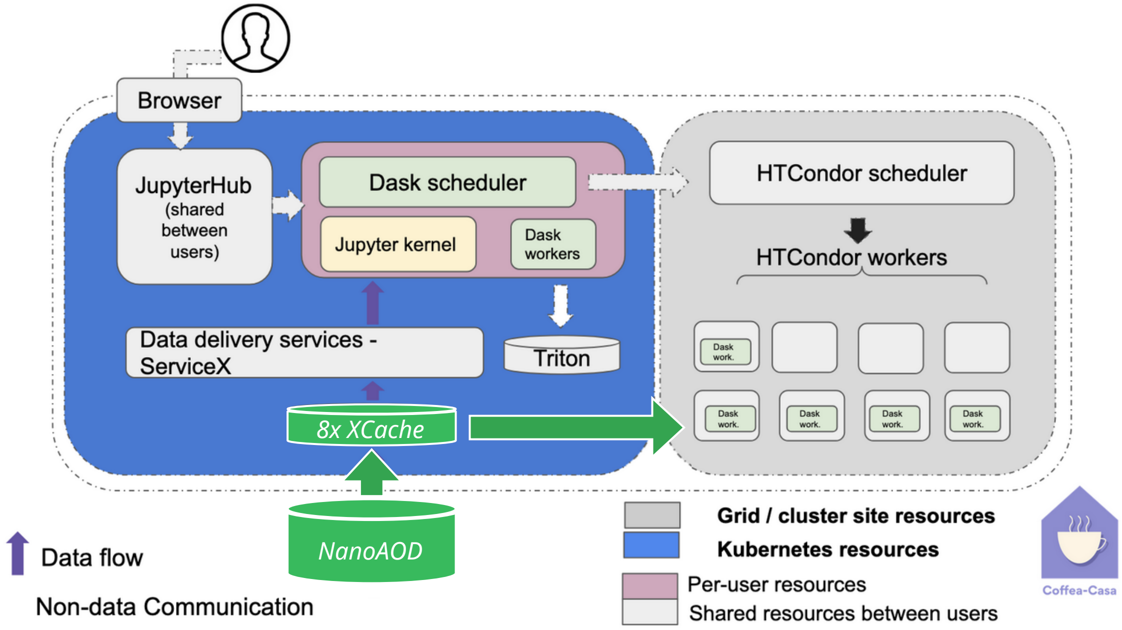
200+ Gbps sustained network throughput

sufficiently many worker nodes to process data see Oksana Shadura's talk

Coffea-casa at Nebraska: 200 Gbps setup

developed in close collaboration

- **R&D prototype of a future Analysis Facility**
  - designed as hybrid setup including Kubernetes and Nebraska CMS Tier-2 / Tier-3 resources



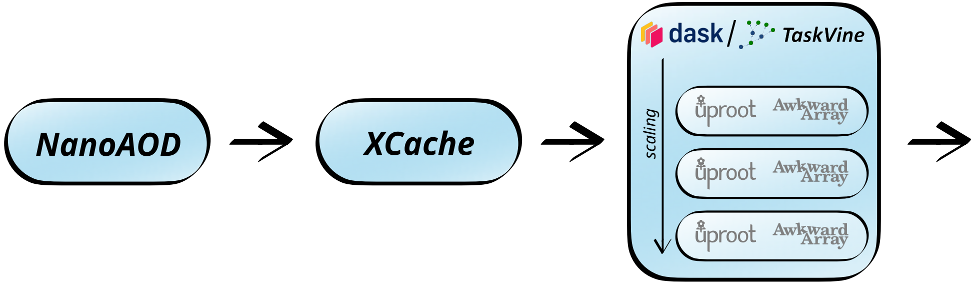
using 8 XCache instances behind 2x100 Gbps uplink each

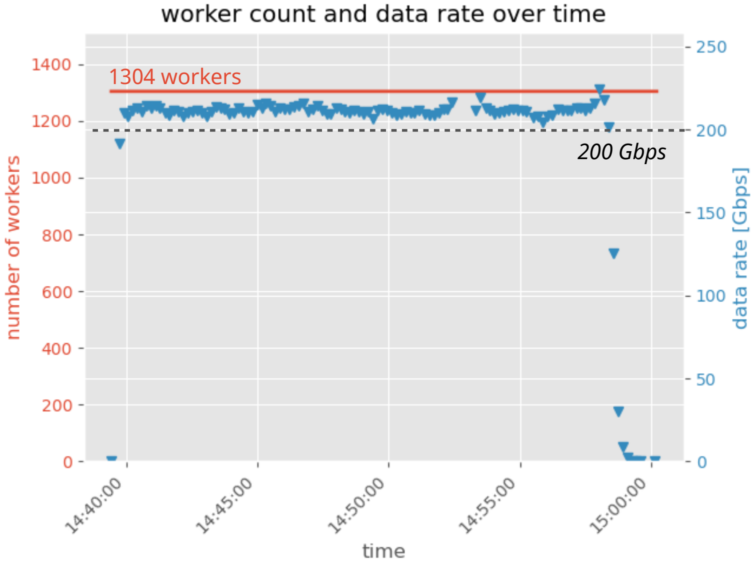
Coffea-casa at Nebraska: measurements

- **200 Gbps sustained throughput of data for physics**
  - scheduling with Dask & TaskVine, scaling with HTCondor & Kubernetes
  - re-compressed NanoAOD (LZMA → ZSTD) for 2.5x event rate increase

details for this example:

- *40 B events, 64k files*
- *1304 workers*
- *32 MHz event rate*
- *data processed (compressed): 30 TB*
- *data processed (uncompressed): 71 TB*





200+ Gbps with Dask + HTCondor

see Rob Gardner's talk

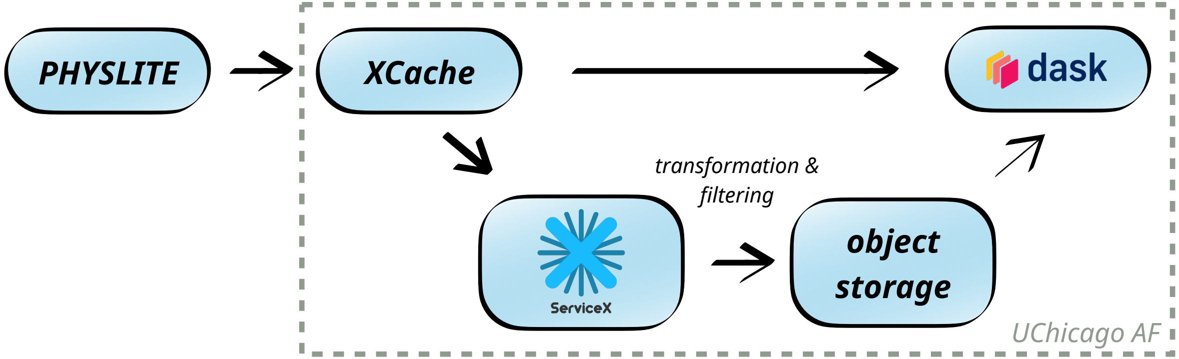
UChicago AF: 200 Gbps setup

developed in
close collaboration

US ATLAS iris hep

- **Production Analysis Facility for ATLAS**
  - built on Kubernetes, partially reconfigured for needs of challenge

- **Two configurations explored** with Kubernetes scaling (HTCondor available)
  - `uproot` scaled with Dask reading from XCache
  - ServiceX as data delivery service writing to object storage

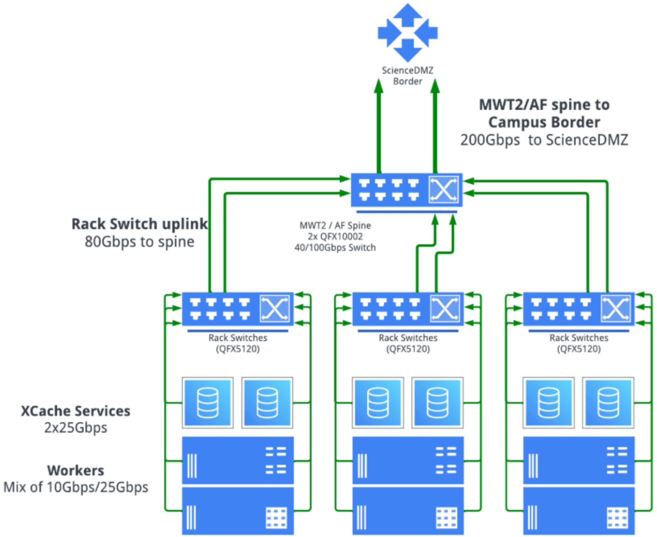


PHYSLITE → XCache → dask

XCache → ServiceX → object storage → dask

transformation & filtering

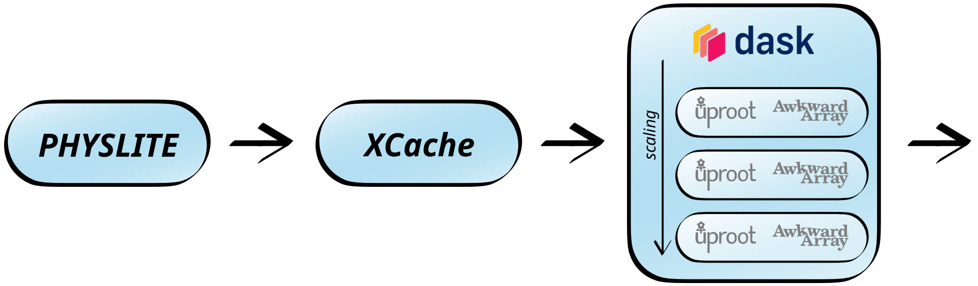
UChicago AF

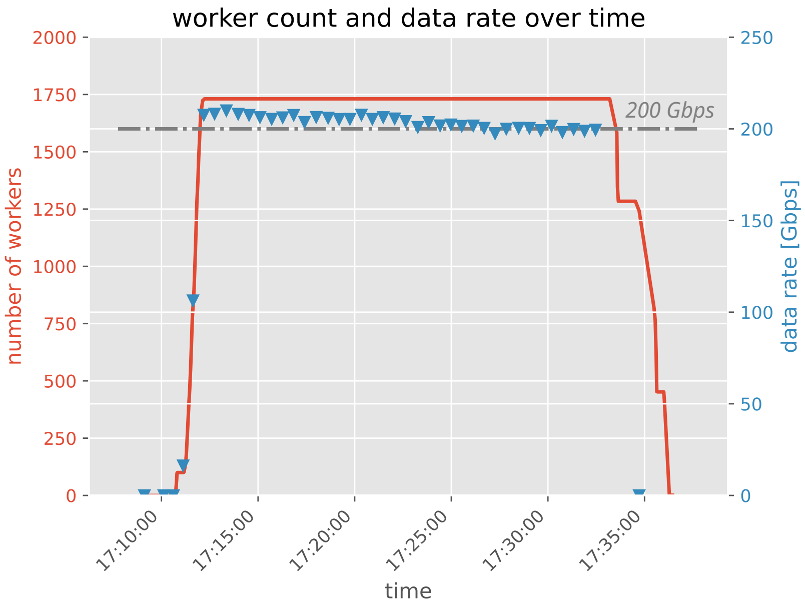


8 XCache instances total,
distributed to maximize bandwidth

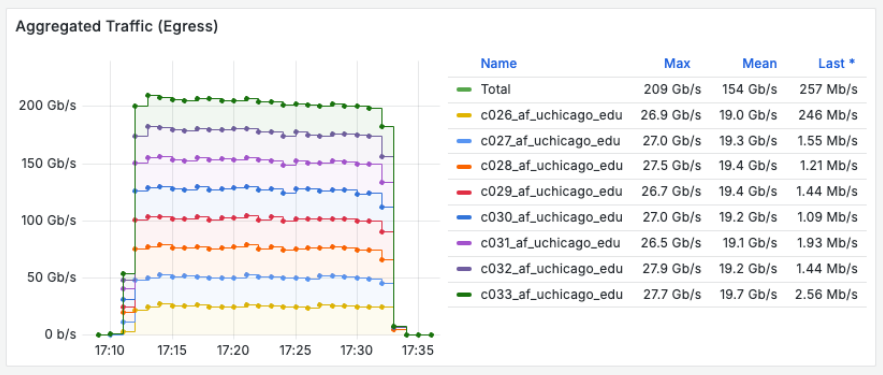
UChicago AF: measurements

- **200 Gbps sustained throughput of data for physics**





network monitoring



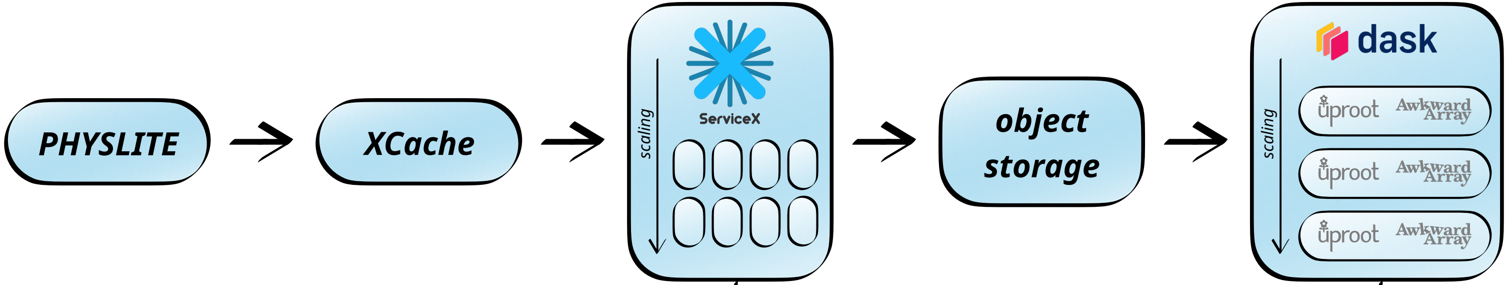
Name	Max	Mean	Last *
Total	209 Gb/s	154 Gb/s	257 Mb/s
c026_af_uchicago_edu	26.9 Gb/s	19.0 Gb/s	246 Mb/s
c027_af_uchicago_edu	27.0 Gb/s	19.3 Gb/s	1.55 Mb/s
c028_af_uchicago_edu	27.5 Gb/s	19.4 Gb/s	1.21 Mb/s
c029_af_uchicago_edu	26.7 Gb/s	19.4 Gb/s	1.44 Mb/s
c030_af_uchicago_edu	27.0 Gb/s	19.2 Gb/s	1.09 Mb/s
c031_af_uchicago_edu	26.5 Gb/s	19.1 Gb/s	1.93 Mb/s
c032_af_uchicago_edu	27.9 Gb/s	19.2 Gb/s	1.44 Mb/s
c033_af_uchicago_edu	27.7 Gb/s	19.7 Gb/s	2.56 Mb/s

more details:

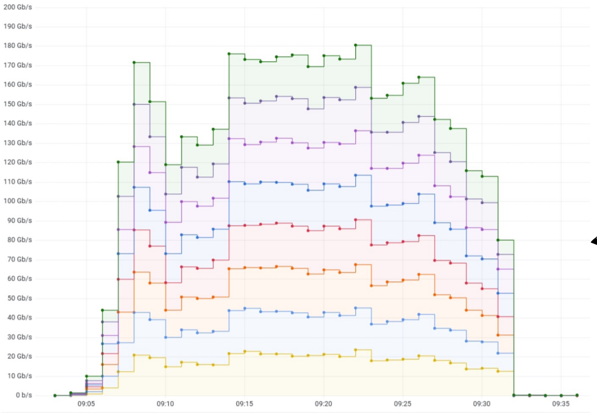
- *32 B events, 190 TB data, 218k files*
- *1739 workers peak*
- *15 MHz event rate, 5–20 kHz per core*
- *200 Gbps throughput sustained*
- *data processed (compressed): 32 TB*
- *data processed (uncompressed): 80 TB*

ServiceX as data delivery service

- **Idea: filter data with ServiceX**, then further process output with Dask
    - rapid turnaround from cached ServiceX output



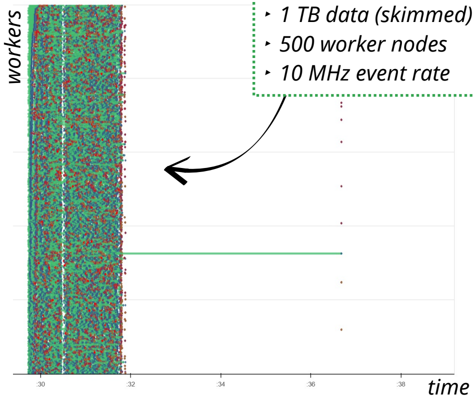
170 Gbps parallel data processing with ServiceX



- *19 B events, 146 TB data*
- *up to 1k pods*
- *10 MHz event rate*

multi-stage processing schema, transparent to users

Dask tasks



- *1 TB data (skimmed)*
- *500 worker nodes*
- *10 MHz event rate*

Towards multi-user interactivity

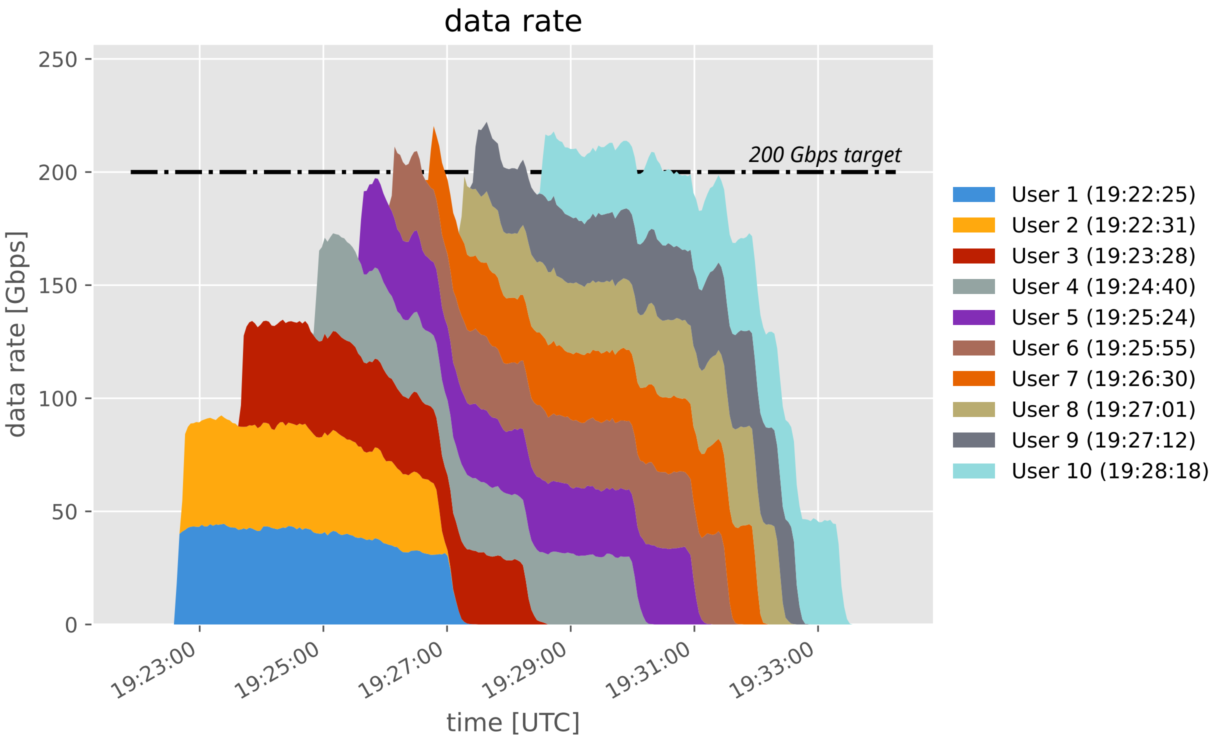
- Analysis Facilities will host **many users** looking to achieve **interactive turnaround simultaneously**
    - ensure scale testing includes number of users

- **Live exercise** at 2024 IRIS-HEP retreat: ***200 Gbps setup with 16 participants***
    - automatic CPU scaling with Dask
    - limited maximum number of cores per user

- Intended as part of a bigger **discussion about fair-share & interactivity**

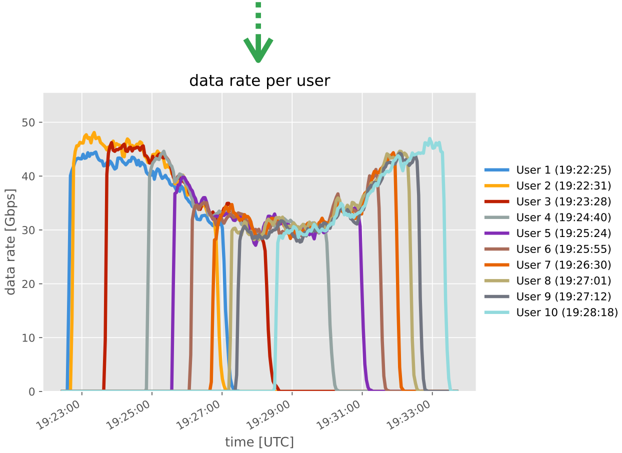
live demonstration with retreat participants

Bandwidth shared between multiple users

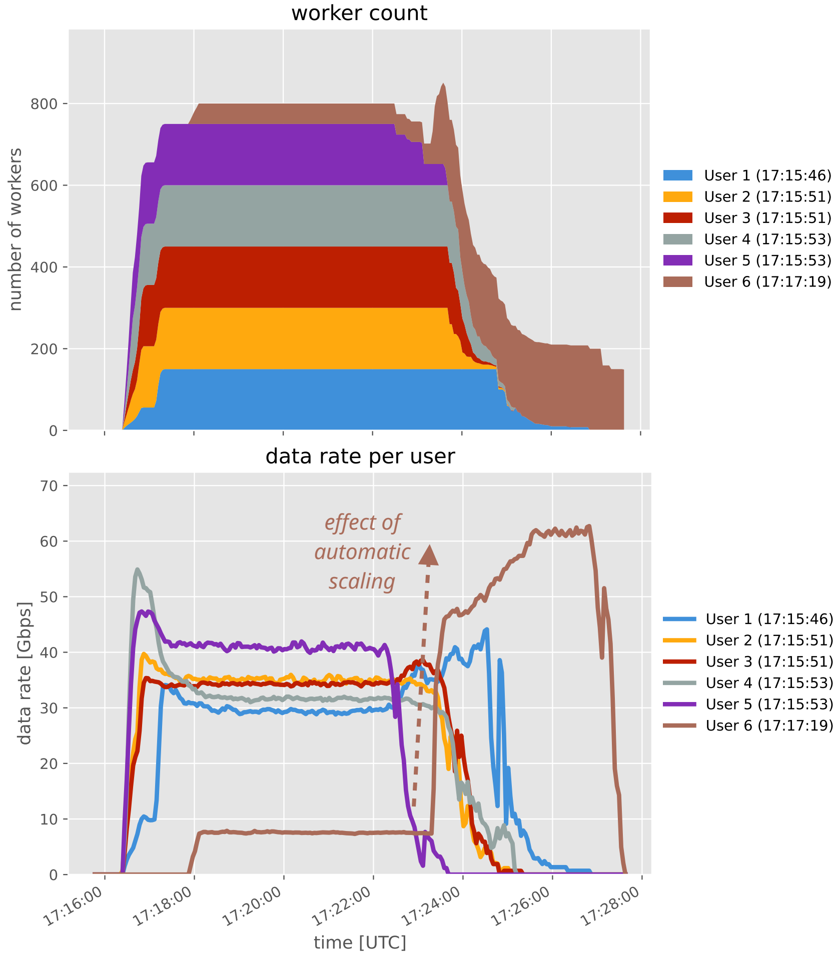


task launch times were randomly distributed to simulate reality of random submissions

- Test with **ten simultaneous users at UChicago**
  - users limited to max 200 cores
- Reached **200 Gbps collectively**
  - network saturation effect visible



Automatic worker scaling



- Test with **six simultaneous users at Nebraska**
- First five users launch at the same time
  - stable parallel processing
- **User 6** receives last available cores, slower initially
  - rapid automatic scaling once more resources become available

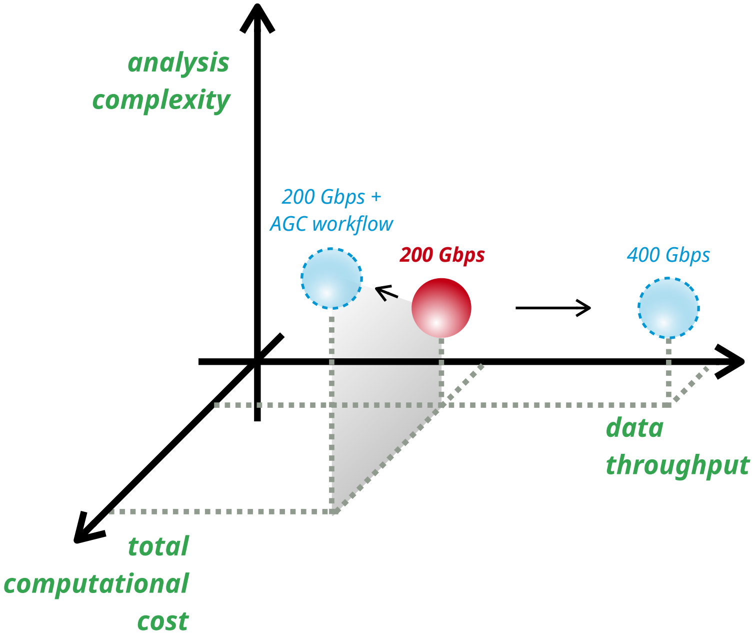
Where to go from here?

Next steps

- **Further explore the parameter space** of HL-LHC analyses
  - extend 200 Gbps setup towards full AGC-type workflow
  - medium term: 400 Gbps exercise

- Further **collaboration with community & knowledge sharing**
  - lessons learned help analyses *already today*

- Help **inform Analysis Facility evolution**



axis closely connected to environmental sustainability

Summary

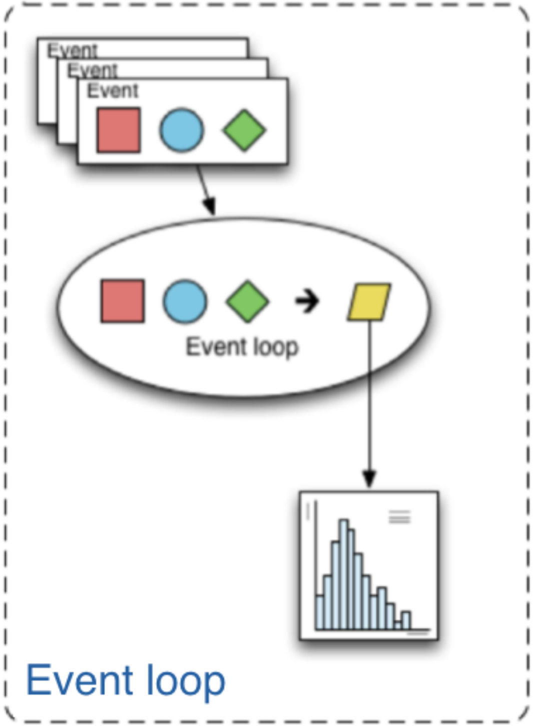
- Developed a **modern analysis pipeline** for efficient HEP data analysis
  - studying & improving performance and usability
  - task graphs with Dask as a central element

- **Successful 200 Gbps Challenge** shows technology readiness, checkpoint towards HL-LHC
  - extremely valuable project to generate feedback and identify potential bottlenecks
  - planned extensions for more realism (Analysis Grand Challenge)

Backup

Columnar analysis / array programming

- In **contrast** to **event loops**, **columnar analysis** processes (chunks of) columns/**arrays-at-a-time** -> e.g. `numpy`
  - makes analysis in Python computationally feasible: optimized implementations of expensive operations

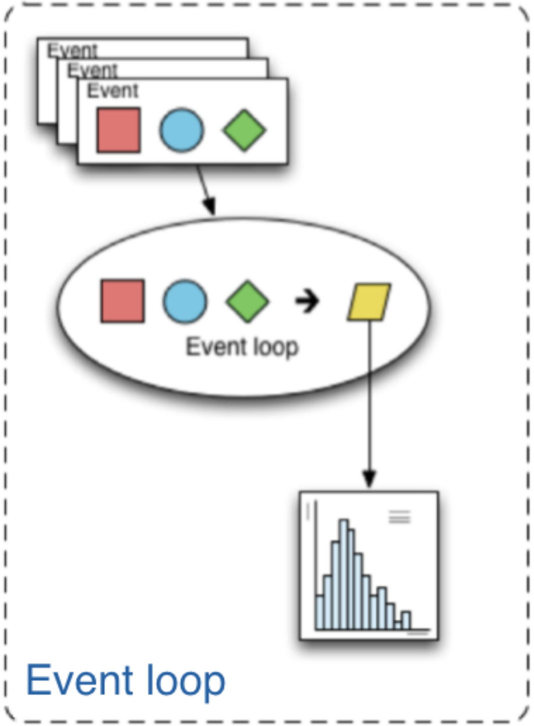


```
for e in events:
     for jet in e.jets:
          if jet.pt > 25:
               hist.fill(jet.pt)
```

https://coffeateam.github.io/coffea/concepts.html

Columnar analysis / array programming

- In **contrast** to **event loops**, **columnar analysis** processes (chunks of) columns/**arrays-at-a-time** -> e.g. numpy
  - makes analysis in Python computationally feasible: optimized implementations of expensive operations

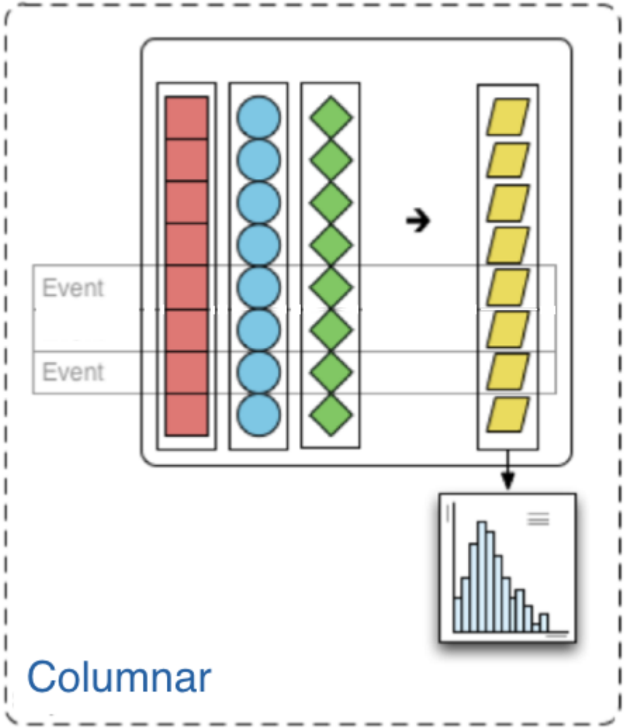


```
for e in events:
     for jet in e.jets:
          if jet.pt > 25:
               hist.fill(jet.pt)
```

different interfaces & mental models

```
mask = events.jets.pt > 25
observable = events.jets.pt[mask]
hist.fill(flatten(observable))
```

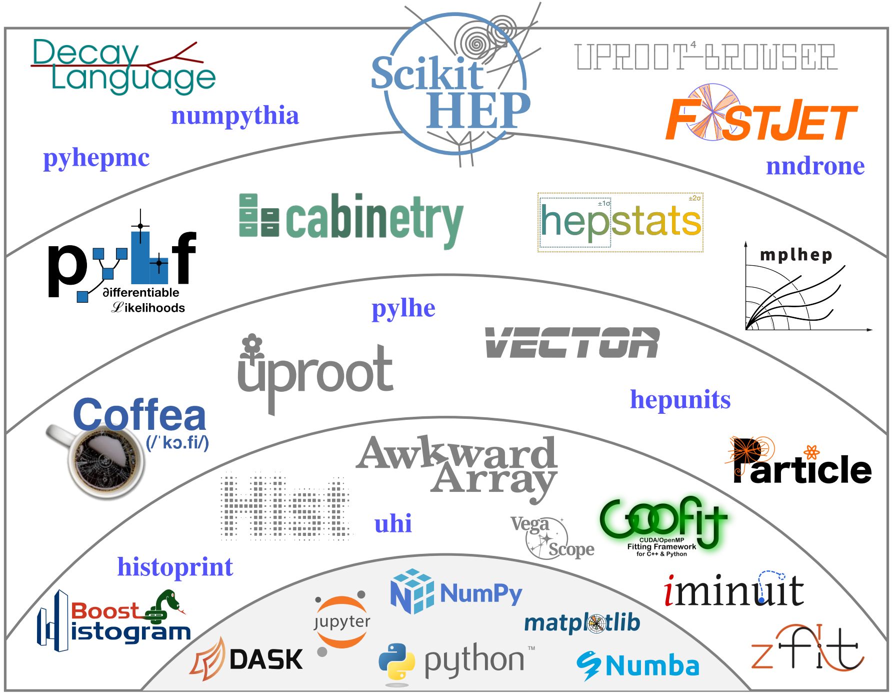
implicit inner loops!



https://coffeateam.github.io/coffea/concepts.html

The HEP Python ecosystem

- **Interoperable HEP Python ecosystem** emerged over past years, building upon & extending **data science tool stack**
    - disclaimer: focusing on scientific HEP Python ecosystem — many things also happening in ROOT!



numpythia

pyhepmc

nndrone

pylhe

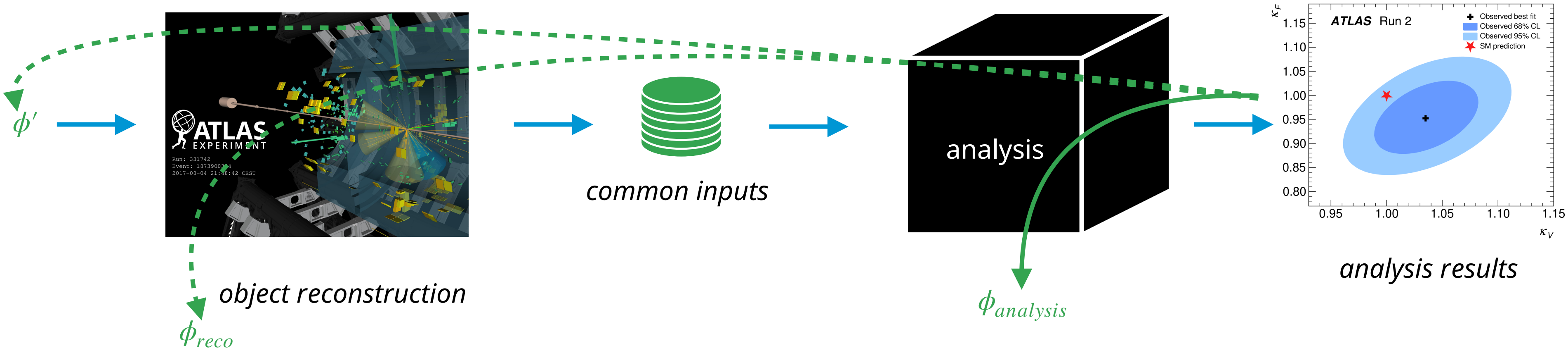
hepunits

uhi

histoprint

Differentiable programming for physics analysis

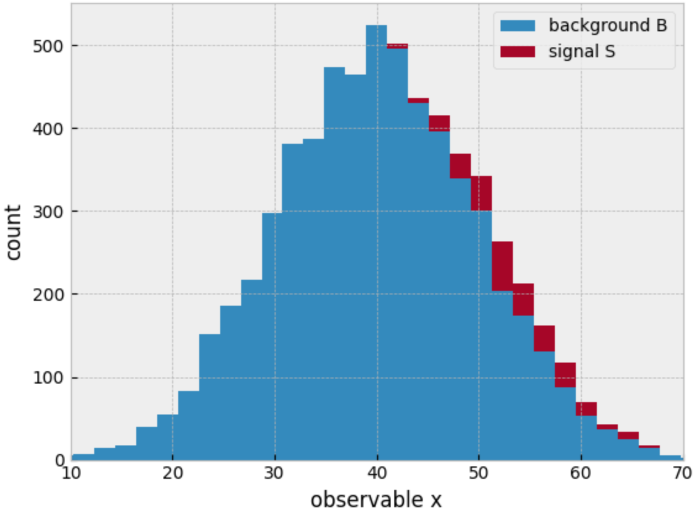
- A differentiable analysis would allow **optimizing physics analysis parameters ϕ via gradient descent**
  - what is the right loss function? can we do this in a manner that is robust to mismodeling?



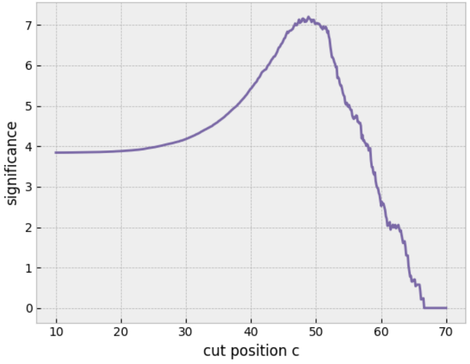
- Exploration of **differentiation of parts of this pipeline** has been ongoing for a while
  - example: pyhf + neos for end-to-end optimized summary statistics arXiv:2203.05570

Challenge: non-differentiable operations

- Physics analysis regularly uses **non-differentiable operations** (cuts, binning, sorting, …)

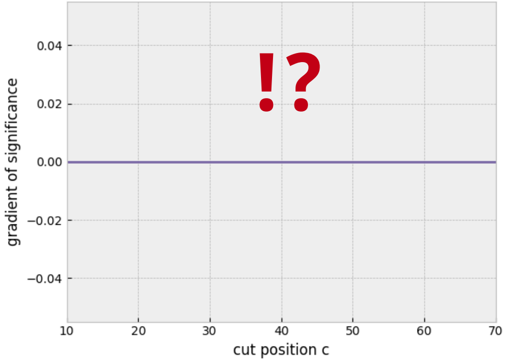


one-bin counting experiment after requiring $x > c$



traditional approach: parameter scan to optimize significance

$\frac{\partial}{\partial c}$ significance

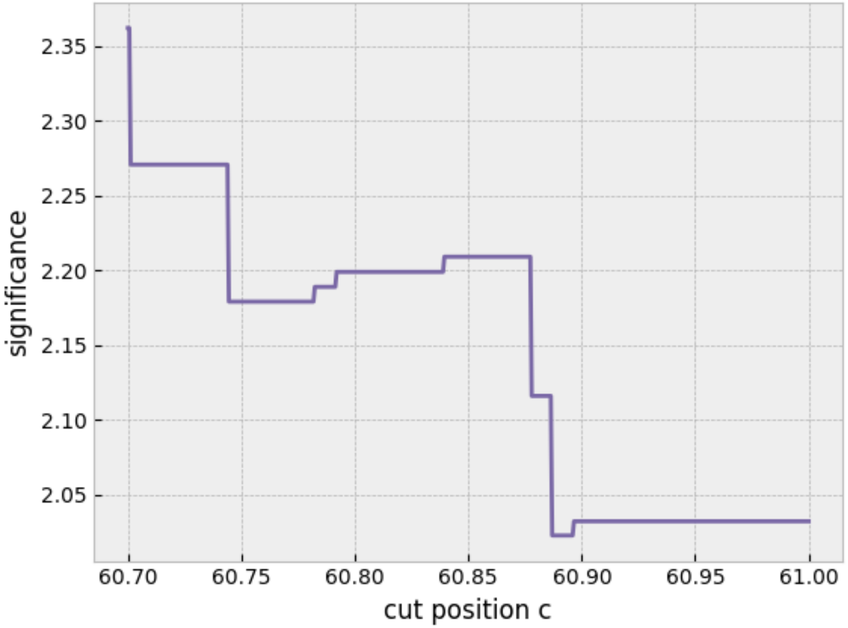


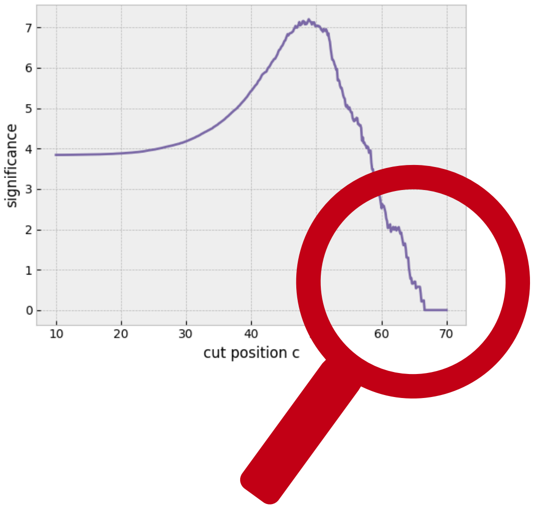
example notebook with these figures

Challenge: non-differentiable operations

- Physics analysis regularly uses **non-differentiable operations** (cuts, binning, sorting, …)

discrete steps as individual events pass / fail cuts



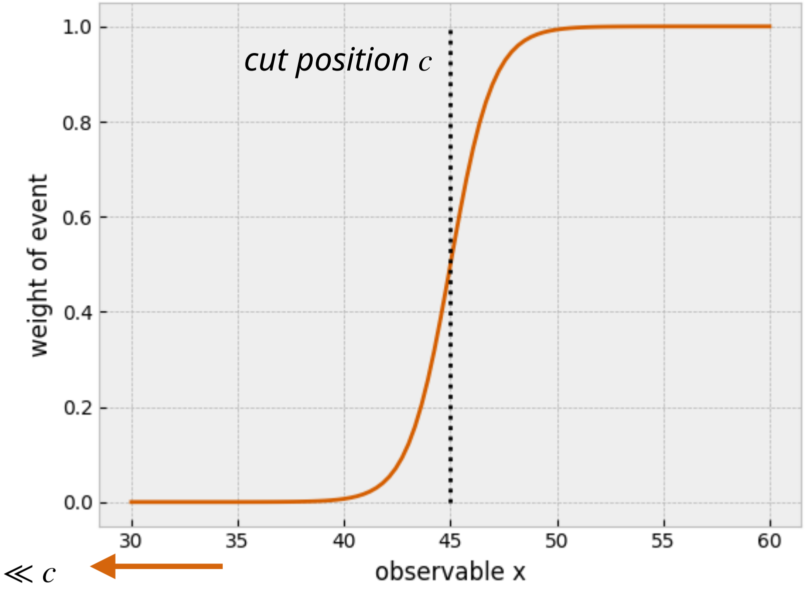


Differentiable cuts

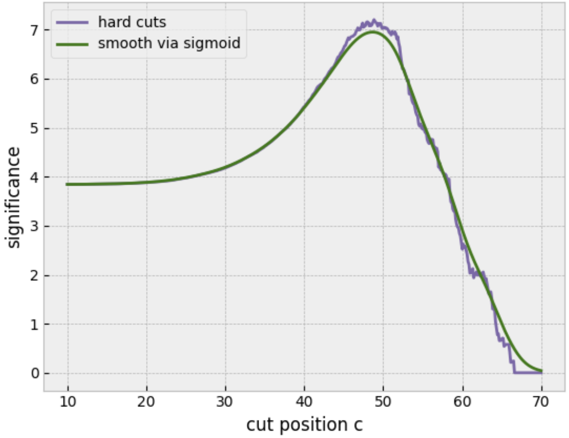
- Remedy the situation by **changing how a “selection cut” acts**

relax the definition of a “cut”:
replace by a weight

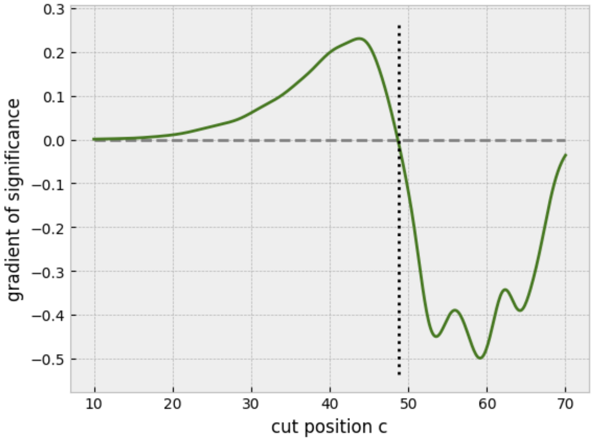
$x \gg c$
weight $\rightarrow 1$



$x \ll c$
weight $\rightarrow 0$

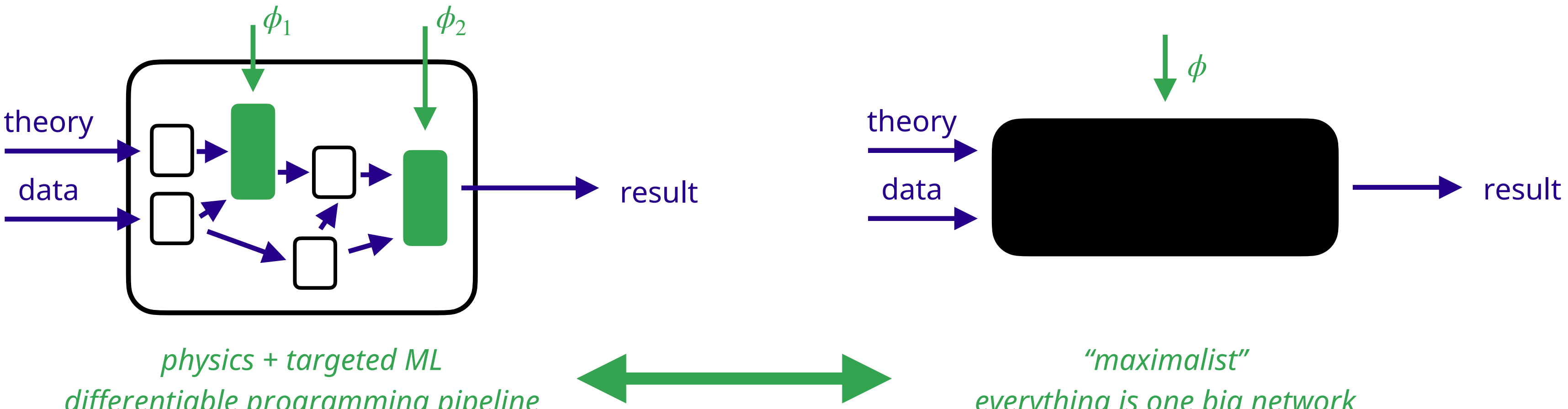


smooth significance estimate + sensible gradient



The future

- **Where is this going?**
  - how much can we gain in sensitivity / how computationally efficient is this?
  - how do we best inject inductive bias / achieve understanding with black box neural networks?



physics + targeted ML
differentiable programming pipeline

“maximalist”
everything is one big network

inspired by: L. Heinrich, ACAT 2024