Run-2 Physics Analysis Workflow using Software for HL-LHC

KyungEon Choi, Peter Onyisi, Marc Tost

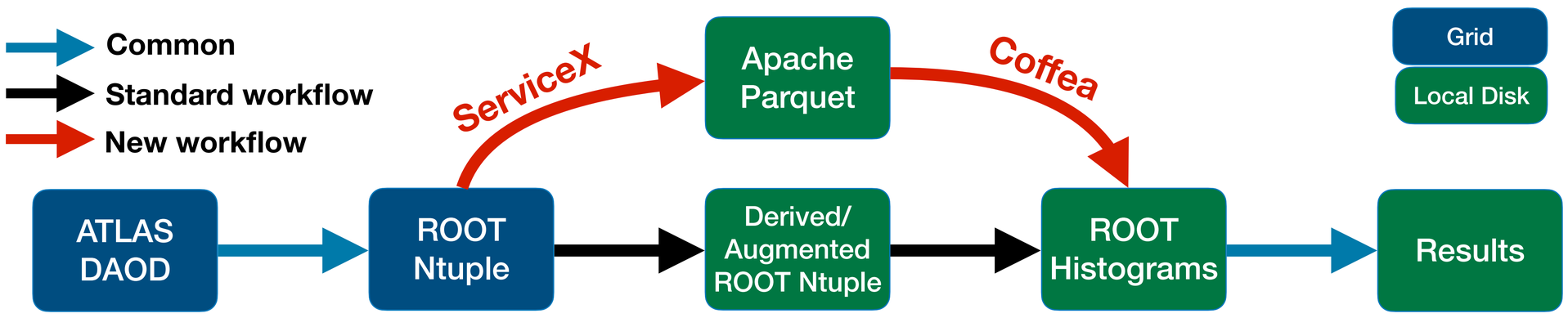
The University of Texas at Austin

Overview

The challenge of handling an order of magnitude more data from the High-Luminosity LHC (HL-LHC) demands novel approaches. Many fascinating software under development to tackle the challenge.

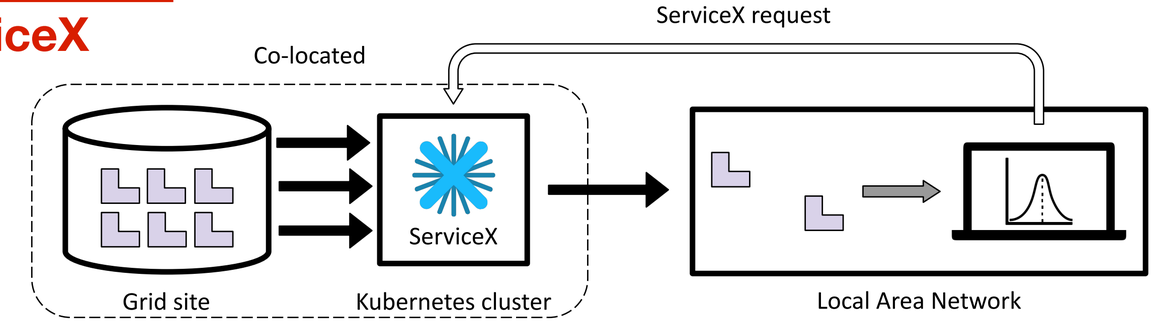
This work utilizes software in scientific python ecosystem, primarily from IRIS-HEP [1], to investigate an alternative workflow for the ongoing ATLAS Run-2 physics analysis.

Analysis workflow



Data access

ServiceX



ServiceX [2] is a scalable HEP event data extraction, transformation, and delivery system.

- HEP event data: various input data formats such as ATLAS xAOD, CMS NanoAOD, ROOT Flat ntuple
- Extraction: user-selected column(s) with event filtering → reduced data over WAN
- Transformation: output in various formats such as awkward arrays, Apache parquets, ROOT ntuple
- Delivery: on-demand, deliver to a user or stream into Analysis Facility
- Scalable: runs on Kubernetes cluster, scales up workers when necessary

ServiceX DataBinder [3] is a python package to make ServiceX data delivery requests and manage ServiceX datasets from a configuration file.

- Multiple samples can be defined in a configuration file
- A sample contains Rucio [4] dataset ID(s), name of tree, selection of columns and events in FuncADL [5] or TCut [6] syntax
- A user may write a configuration file for whole analysis, or per region, or only for machine learning
- ServiceX supports local data cache → only new/modified triggers ServiceX request

Example ServiceX DataBinder configuration

```
General:
  ServiceXBackendName: uproot
  OutputDirectory: /path/to/output
  OutputFormat: parquet

Sample:
  - Name: ttH
    RucioDID: user.kchoi:user.kchoi.sampleA,
              user.kchoi:user.kchoi.sampleB
    Tree: nominal
    FuncADL: "Select(lambda event: {'jet_e': event.jet_e})"
  - Name: ttW
    RucioDID: user.kchoi:user.kchoi.sampleC
    Tree: nominal
    Filter: n_jet > 5
    Columns: jet_e, jet_pt
```

How to run ServiceX DataBinder

```
from servicex_databinder import DataBinder
sx_db = DataBinder('<CONFIG>.yml')
out = sx_db.deliver()
```

Columnar analysis

Coffea

Coffea [7] is a python package for HEP experiment analysis in python ecosystem.

- Makes use of uproot [8] and awkward array [9] to provide an array-based analysis
- Very easy to scale horizontally by making use of modern big-data technologies such as Apache Spark and Dask
- Supports various input data formats which include Apache parquet and ROOT ntuple
- Features histogramming and plotting tools

Application to analysis

Start point

- 5 TB of ROOT ntuples on the grid
- More than 400 Rucio datasets

End point

- ROOT histograms for the subsequent statistical analysis
- Nice tools such as cabinetry [10] and pyhf [11] in python ecosystem allow to perform full analysis in python, but we go until histogram for this work

ServiceX + Coffea workflow

1. Prepare ServiceX DataBinder configuration file(s) and deliver files in parquet
   - Delivery of ~30 columns from >400 Rucio datasets without filtering takes about 20 mins and parquet files amounts to about 3 GB (single tree only)
   - Size of parquet files reduces to 70 MB when selections for the certain control region are applied (single tree only)
   - Deliver minimal data since only new/modified will trigger new ServiceX request
2. Train neural network using ServiceX delivered parquet data and then implement into coffea
3. Run coffea to produce histograms
   - Run over 3 GB of parquets and produce histograms takes about 1 min using 30 workers
   - Viable to run on laptop

Standard workflow

1. C++ based ROOT event loop to produce slimmed/skimmed/augmented ROOT ntuple.
   - Grid job takes about 1 day
   - The size of derived ROOT ntuple still large since any update requires new production
2. C++ based ROOT event loop to produce histograms.

Lessons

- Need to be familiarized with FuncADL, Awkward array, and other python tools
- Sometimes there are too many ways to solve a problem
- Mostly done in Jupyter notebook, but not good for collaboration

References

[1] https://iris-hep.org
[2] https://servicex.readthedocs.io
[3] https://github.com/kyungeonchoi/ServiceXDataBinder
[4] https://rucio.readthedocs.io
[5] https://github.com/iris-hep/func_adl_servicex
[6] https://github.com/ssl-hep/TCutToQastleWrapper
[7] https://github.com/CoffeaTeam/coffea
[8] https://uproot.readthedocs.io
[9] https://awkward-array.org
[10] https://github.com/scikit-hep/cabinetry
[11] https://github.com/scikit-hep/pyhf

ACAT 2021