Study of columnar data analysis methods to complete a Run 2 ATLAS analysis

Marc Tost, Peter Onyisi, KyungEon Choi – The University of Texas at Austin

Introduction

- Problem:
  - Running an ATLAS analysis typically requires very specific environment:
    - Access to centralized packages
    - Large local disk for downloading ntuples
    - Managed batch computation (HTCondor, grid access)
  - Larger datasets and more computationally expensive processes increase runtime and limit users to specific facilities and workflows
- Solution:
  - Using pythonic packages, a full ATLAS analysis can be run nearly anywhere, including a supercomputing site
- Testing with FCNC t -> Hq multilepton analysis

Triton Inference Server

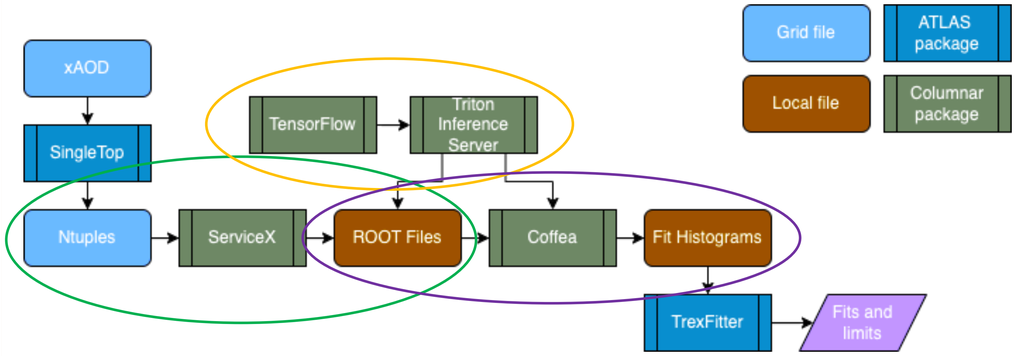
- Machine learning instancing without installing or deploying specific ML packages [6]
- Remote access to GPU!
- Built-in data management achieves maximal GPU use without user input

Parallelization

- Coffea workers easily parallelized with python multithreading [2]
- Additional control can be achieved with Dask [4]
  - Provides active management of workers and computational distribution
  - Dask-MPI provides interface with more complex environments, like supercomputers

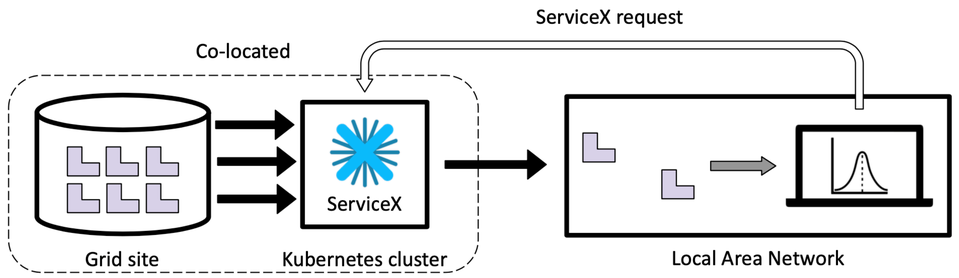
Analysis workflow

1. ATLAS xAOD files are converted to flat ntuple, delivered on grid [7]
2. ServiceX accesses ntuples, returns user-specified columns. 40k ROOT files (420 GB) are delivered
3. Coffea is used to analyze data and run calculations
4. Results are saved as ROOT histograms.
5. Final fits and limits are extracted using TrexFitter [8]



ServiceX

- Data extraction, transformation and delivery tool with a wide range of HEP applications [1]
- Extremely portable, especially with use of Databinder [3]
- Ready to deliver data with 3 simple scripts and two packages installed with pip
- Works out of the box at all tested locations

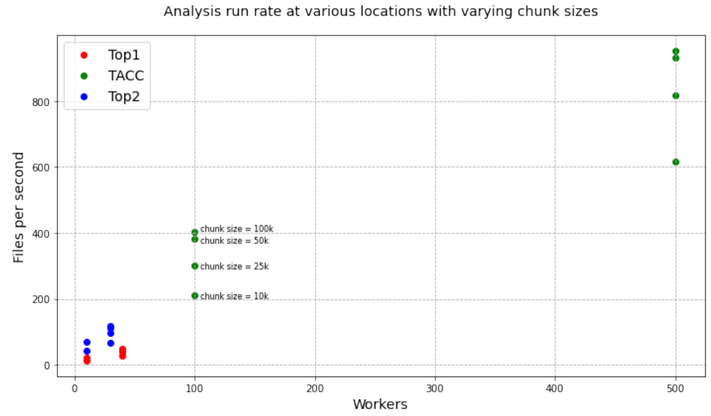


Learn more about ServiceX!

Supercomputing

- Supercomputers have massive computing power, but are traditionally hostile to HEP analyses
  - Minimal packages installed, no access to ATLAS analysis tools
  - MPI architectures for distribution, usually incompatible with analysis techniques
- Testing at TACC (Texas Advanced Computing Center) [5]
  - Running at Lonestar 6 (intel machine)
  - Access through Jupyter interface, very convenient for pythonic interface
  - Scaling up to 500 cores
    - Another order of magnitude is possible!
    - Currently limited by opening/closing many files – more tests soon!

Analysis run rate at various locations with varying chunk sizes



- Setup:
  - Packages (~15) installed with pip
  - ServiceX Databinder worked out of box, transformed and delivered nominal dataset in less than 1 hour
  - Coffea worked with 3 lines of code added, less than an hour of effort
- Full ATLAS analysis running on supercomputer with less than an afternoon's work!

References:

[1] https://servicex.readthedocs.io
[2] https://github.com/CoffeaTeam/coffea
[3] https://github.com/kyungeonchoi/ServiceXDataBinder
[4] https://dask.org
[5] https://tacc.utexas.edu
[6] https://developer.nvidia.com/triton-inference-server
[7] https://gitlab.cern.ch/atlasphys-top
[8] https://trexfitter-docs.web.cern.ch/trexfitter-docs/