AGC-column-joining: On-demand column-joining of input data for the Analysis Grand Challenge demonstrator analysis.

.

Applicant Patrikas Rudaitis
Mentor Nick Manganelli
Proposed project duration 8 weeks.
Proposed start date 2025/07/01.

1 Introduction

The upcoming LH-LHC[1] upgrade will usher in higher luminosity measurements which could lead to discovery of new physics. However, these hardware upgrades will bring with it challenges to computing and storage of the data. NanoAOD - which is an analysis-ready data format, was created to ease the requirements for computing and storage, but this format does not cover the needs for all analysis in CMS. Sometimes more data is required, data that is only accessible on larger data formats (miniAOD, AOD). The usual analysis workflow creates redundant copies of data which is an inefficient use of disk space - a problem that will only be exacerbated with the increase of the volume of recorded data.

A way to reduce disk usage would be to create a module that eliminates the need for unnecessary copying. This module must be able to quickly join nanoAOD data with some auxiliary data, needs to be stable and easily scalable. It also needs to support the joining of data with little manual derivation. A prototype of such a module is currently being developed, therefore this project proposal will focus on the prototype's integration with the AGC(Analysis Grand Challenge)[2] and additional extensions to the prototype.

2 Proposal

The main goal of this project would be the integration of the existing column-joining prototype with the AGC for the purpose of benchmarking and stress testing the prototype. This will necessitate the study of the existing prototype and the AGC as that would allow me to familiarize myself with the project - ideally, this step should take as little time as possible. In addition to the main goal I would also be focusing on an integration with the 200 Gbps challenge[3], prototyping RNTuple[4] integration and improving generalizability. I would also test out alternative join engines if enough time is available and the necessity of the tests are sufficient. Another possible goal would be to integrate column joining with the ATLAS part of the AGC.

3 Goals

1. Integration of the column-joining prototype with the AGC. [CORE GOAL]
   (a) Study the existing column-joining prototype as well as the AGC itself to identify the best way to integrate. [time frame: up to a week.]

   (b) Integrate the column-joining prototype with the CMS AGC along with testing of the integration. [time frame: this would take up to bulk of the allotted time, three weeks minimum.]
   (c) Stress-test the integration with the 200 Gbps challenge. [time frame: up to a week.]
   (d) If time permits, integrate the column-joining prototype with ATLAS AGC.

2. Prototyping RNtuple integration. [Secondary GOAL]
   (a) Evaluate the best method to integrate RNTuple with the AGC. [time frame: less than a week.]
   (b) Attempt integration of RNTuple with the AGC. This milestone requires that the integration of RNTuple into uproot/coffea becomes sufficiently advanced. If it becomes clear that integration will not succeed, focus will be shifted to other goals. [time frame: two weeks.]
   (c) Benchmark the integration. [time frame: less than a week.]

3. Testing out alternatives to Trino[5]. [Tertiary GOAL] [time frame: one week.]

4 References

[1] CERN's website, High-Luminosity LHC, `https://home.cern/science/accelerators/high-luminosity-lhc`

[2] IRIS-HEP website, The Analysis Grand Challenge, `https://iris-hep.org/projects/agc.html`

[3] IRIS-HEP website, 200 Gbps Challenge, `https://iris-hep.org/projects/200gbps.html`

[4] ROOT Refrence guide, RNTuple Introduction, `https://root.cern/doc/master/md_tree_2ntuple_2v7_2doc_2README.html`

[5] Trino documentation, `https://trino.io/docs/current/`