Implementing ServiceX data as a source for ROOT's RDataFrame

Nicholas Decheine
Department of Physics, University of Wisconsin–Madison
Mentor: Gordon Watts

Duration: June 2021 - August 2021

ServiceX is a smart data delivery service used by physicists to retrieve data subsets for analysis. It is an experiment-agnostic service that enables on-demand columnar data delivery tailored for high performance, array-based analysis. It supports ATLAS xAOD, CMS NanoAOD, and flat n-tuple datasets. A user requests data subsets and transformations, such as filters or computations, and it returns in a columnar format. ServiceX has native support for specifying the output of a request as a ROOT TTree.

ROOT is a leading data analysis framework used by high energy physicists and data scientists. RDataFrame is ROOT's declarative analysis interface. It offers a high level interface for analysis of data stored in formats such as TTree's and CSV files, and other custom formats can be created. Programs utilizing RDataFrame construct data frame objects from a given input data set.

The data frame can be transformed by applying filters to specific rows, creating new columns that store computations performed for a row, among other analysis capabilities. Finally, results can be produced via data aggregation into meaningful analysis, such as histograms and plots. RDataFrame even supports multithreaded operations with native support for ROOT's implicit multi-threading which provides an internal parallelization mechanism.

The endeavor at hand is to have the framework that takes ServiceX request information from the user and returns an analysis-ready RDataFrame instance based on the ServiceX request. From the user's point of view, the process is aimed to be as simple as possible. This will involve understanding the ServiceX API, the RDataFrame structure, and learning how to implement web API calls using C++. The software will need to interpret and fulfil user-provided ServiceX queries and make the necessary requests before processing the result into an RDataFrame. There will be a working and documented demo that guides a user through the usage of this software, taking them from writing the query to the stage were they can make operations on the resulting RDataFrame. The demo dataset will be CMS Higgs event data from CERN's Open Data collection.

This program allows a researcher to have a more streamlined data acquisition experience, piping cloud event data straight into their ROOT analysis environment.This project will be conducted remotely under the guidance of Dr. Gordon Watts at the University of Washington, with the objective of creating a program that creates an RDataFrame object with queried ServiceX grid data by August. If this is completed before the project end date, a stretch goal objective is to incorporate ROOT's native multithreading capabilities into the software to speed up performance. The deliverables of this project will be the C++ software that does all the work, and a documented demo of the software utilizing the Open Data dataset chosen. The timeline of the project is given on the next page.

Timeline

Week	Goal
1	Project setup. Learn and understand RDataFrame and the data sources at each step of the pipeline. Set up ServiceX for use by obtaining necessary credentials.
2	Learn how to implement web APIs with C++ - submitting and retrieving data.
3	Design and discussion of C++ implementation of ServiceX web API calls to fetch grid data.
4-5	Design and implement a ServiceX query input system readable by the software.
6	Writing tests for ServiceX queries and their expected ROOT file outputs
7-8	Implementation of ServiceX TTree output reader for processing the input for RDataFrame; make dummy input ROOT file for testing data source.
9	Integration of features into standalone framework.
10-11	Create a demo with documentation guiding a user through the usage of the data stream system, using Higgs-Discovery demo data as the dataset
12	Buffer period for bug squashing, finishing the demo, and/or completion of stretch goals, like multithreading capabilities.

Academic Workload: During this summer term, I will be taking only one course, Algorithms, to finish my computer science degree at the University of Wisconsin-Madison. This project will be my full-time commitment.