ACAT 2022

Contribution ID: **151**

Type: **Oral**

Using a DSL to read ROOT TTrees faster in Uproot

Wednesday 26 October 2022 14:55 (20 minutes)

Uproot reads ROOT TTrees using pure Python. For numerical and (singly) jagged arrays, this is fast because a whole block of data can be interpreted as an array without modifying the data. For other cases, such as arrays of std::vector<std::vector<float», numerical data are interleaved with structure, and the only way to deserialize them is with a sequential algorithm. When written in Python, such algorithms are very slow.

We solve this problem by writing the same logic in a language that can be executed quickly. AwkwardForth is a Domain Specific Language (DSL), based on Standard Forth with I/O extensions for making Awkward Arrays, and it JIT-compiles to a fast virtual machine without requiring LLVM as a dependency. We generate code as late as possible to take advantage of optimization opportunities. All ROOT types previously implemented with Python are being converted to AwkwardForth.

Double and triple-jagged arrays have already been implemented and are 400× faster in AwkwardForth than in Python, with multithreaded scaling up to 1 second/GB because AwkwardForth releases the Python GIL. In this talk, we describe design aspects, performance studies, and future directions in accelerating Uproot with AwkwardForth.

Significance

This talk presents an implementation of the acceleration anticipated in the talk and paper referenced below. Previously, the I/O speed was measured in a mocked-up (but realistic) test, now it is implemented in Uproot in a way that will be used in production, in Uproot version 5. The observe the same magnitude of speed-up (with respect to the state-of-the-art Uproot 4) as in the previous talk and paper.

References

https://indico.cern.ch/event/948465/contributions/4324131/ (vCHEP 2021)
https://inspirehep.net/literature/1849024

Experiment context, if any

IRIS-HEP

Primary authors: ROY, Aryan; PIVARSKI, Jim (Princeton University)

Presenter: ROY, Aryan

Session Classification: Track 1: Computing Technology for Physics Research

Track Classification: Track 1: Computing Technology for Physics Research