



Contribution ID: 618 Contribution code: **contribution ID 618**

Type: **Oral**

Lessons learned in Python-C++ integration

Thursday, December 2, 2021 11:20 AM (20 minutes)

Awkward Array 0.x was written entirely in Python, and Awkward Array 1.x was a fresh rewrite with a C++ core and a Python interface. Ironically, the Awkward Array 2.x project is translating most of that core back into Python (leaving the interface untouched). This is because we discovered surprising and subtle issues in Python-C++ integration that can be avoided with a more minimal coupling: we can still put performance-critical code in C++, but also benefit by minimizing the interface between the two languages.

This talk is intended to share what we learned from our experiences: design choices that look innocent but can cause issues several steps later, often only in the context of real applications. The points to be presented are (1) memory management: although Python references can be glued to `std::shared_ptr`, cycles through C++ are invisible to Python's garbage collector and can arise in subtle ways, (2) C++ standard library types are not a portable runtime interface, owing to ABI differences, and (3) tracers, at the heart of Python libraries like Dask and JAX, can only be fully leveraged if black-box calls out of Python use basic, universally recognized types: flat arrays, not objects.

The goal of this talk is to call out these issues so that other projects mixing Python and C++ can avoid them in the design stage.

Significance

This talk is not aimed at potential users of Awkward Array, but developers working on other projects that need to mix Python and C++. This need is increasingly relevant as more front-ends use Python, but large-scale processing still needs to have high performance.

References

The closest match to this talk's content can be found in an IRIS-HEP Analysis Systems group meeting: <https://indico.cern.ch/event/1032972>

But this talk would be for a wider audience, presenting these issues in a more general way: more like "how to" tips than specific project plans.

Speaker time zone

No preference

Author: PIVARSKI, Jim (Princeton University)

Presenter: PIVARSKI, Jim (Princeton University)

Session Classification: Track 1: Computing Technology for Physics Research

Track Classification: Track 1: Computing Technology for Physics Research