



Contribution ID: 78

Type: Oral

Vectorized, imperative, and declarative processing of Awkward Arrays

Thursday, November 7, 2019 3:00 PM (15 minutes)

Over the past two years, the uproot library has become widely adopted among particle physicists doing analysis in Python. Rather than presenting an event model, uproot gives the user an array for each particle attribute. In case of multiple particles per event, this array is jagged: an array of unequal-length subarrays. Data structures and operations for manipulating jagged arrays are provided by the awkward-array library, which also includes array types for nested structures, nullability, and heterogeneity.

The primary mode of awkward-array manipulation is vectorized in the sense of a Single Python Instruction on Multiple Data (“virtual machine SIMD”). Many implementations of these high-level instructions are also vectorized in the hardware sense. But whereas the implicit loops of vectorized instructions make some calculations easier, they make others harder, especially algorithms that iterate until a convergence condition is met.

To support algorithms with explicit for loops, we have extended awkward-array to work with Numba, a popular Python JIT-compiler. Vectorized and imperative programming styles may now be used interchangeably on the same awkward-array structures, with no loss of performance.

These two programming styles, column-first vectorized and row-first imperative, differ in their order of execution. We will also show progress on a declarative interface, which doesn’t specify the execution order, allowing the implementation to optimize it independently of the content of the physics analysis itself.

Consider for promotion

Yes

Primary authors: PIVARSKI, Jim (Princeton University); LANGE, David (Princeton University (US)); ELMER, Peter (Princeton University (US))

Presenter: PIVARSKI, Jim (Princeton University)

Session Classification: Track 5 – Software Development

Track Classification: Track 5 – Software Development