Accelerating Awkward Array Builders

Manasvi Goyal

Delhi Technological University, India

Proposed project timeline: May, 2022 - July, 2022 (3 months)

Mentors: Ianna Osborne and Jim Pivarski

Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, to manipulate JSON-like data using NumPy-like idioms. Awkward Arrays provide a more concise, faster and memory efficient alternative to the equivalent Python expression.

I will be starting the project from May 9, 2022 and will be working on it for the next 12 weeks till 31st of July. I will be working on accelerating the array builders and will be using C++ as the primary language.

The project would concentrate on improving the performance of the builders of Awkward Arrays by exploiting different techniques including Just-in-time compilation (JIT).

JIT compilation is a method for improving the performance of interpreted programs. During execution the Python program may be compiled into C/C++ native code to improve its performance. It can be accelerated by generating code specific to each case. Cppyy, an automatic, run-time, Python-C++ binding generator can be used for calling C++ from Python and Python from C++. Run-time generation enables detailed specialization for higher performance, lazy loading for reduced memory use in large scale projects

One task would be making the LayoutBuilder take advantage of a JIT compiler to become as fast as specialised output.

Another, related task would be optimising GrowableBuffer and Forth's OutputBuffers.

The GrowableBuffer is used by the ArrayBuilder to produce “snapshots” of accumulated data by converting them into Awkward Arrays. This is a copy operation, so the buffer growth does not need to be contiguous. The copy can be combined with concatenation. This should provide a significant gain in speed and optimise memory allocation.

This project will prove to be useful in accelerating the performance of the Awkward Array Builders by preventing memory leaks, optimised allocation of memory and improving the speeds.

Milestones

The following are the estimated weekly milestones of the project for the duration of 12 weeks (May - July, 2022) -

1. Week 1

Getting familiar with Awkward Array codebase. Getting a deeper understanding of different C++ concepts relevant to the project.

Create the first PR for the project. Establish a baseline with performance tests and create a table to store the outputs from the tests.

2. Week 2

Complete Growable Buffer : Convert the buffer from contiguous to discontiguous by combining the copy with concatenation to get a significant gain in speed and optimise memory allocation. Run tests to compare the performance after making the changes with the baseline performance.

3. Week 3

Complete Forth's OutputBuffers - Convert the buffer from contiguous to discontiguous and run tests to compare the performance after making the changes with the baseline performance.

4. Week 4 - 5

Getting familiar with cppyy (Automatic Python-C++ bindings). Start working on LayoutBuilder to implement JIT techniques to improve the performance of interpreted programs by generating code specific to each case.

Work on FormBuilder, EmptyArrayBuilder and NumpyArrayBuilder

5. Week 6 - 7

Implement JIT techniques in LayoutBuilder : ListArrayBuilder, RecordArrayBuilder, ListOffsetArrayBuilder, RegularArrayBuilder

6. Week 8 - 9

Implement JIT techniques in LayoutBuilder : UnmaskedArrayBuilder, BitMaskedArrayBuilder, ByteMaskedArrayBuilder, UnionArrayBuilder, IndexedArrayBuilder, IndexedOptionArrayBuilder.

7. Week 10 - 11

The first week will be kept as a buffer to wrap up any incomplete task from the previous weeks and to start running the tests on the improved LayoutBuilder code.

8. Week 12

Documentation of the changes made to the project along with the tools and techniques used. Preparation of the final project presentation.