

Automating Awkward Array Testing

Santam Roy Choudhury

National Institute of Technology, Durgapur

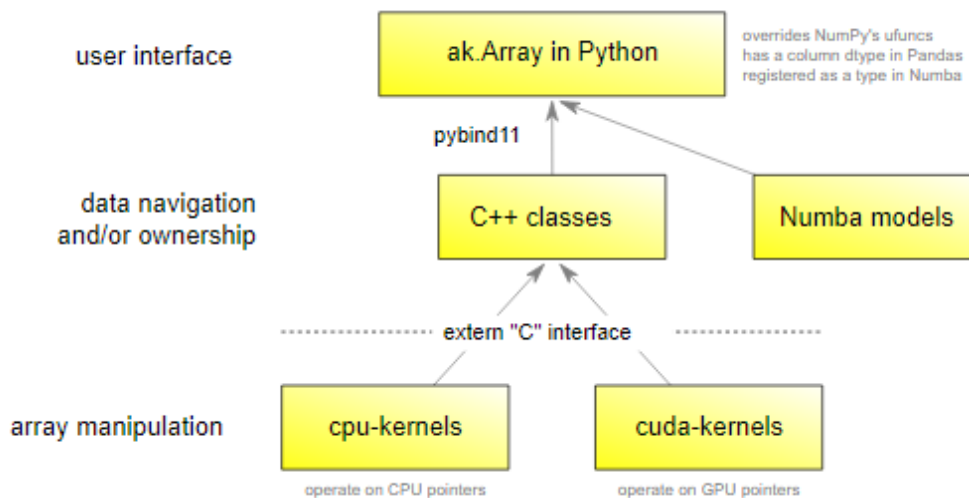
Duration- June,2021 to August,2021 (3 Months)

Mentors- Ianna Osborne and Jim Pivarski

Awkward Array is a popular library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, using NumPy-like idioms.

I will be starting my project work from the 1st of June and will work on it for the next 12 weeks till the end of August. I would be working on writing and testing the kernel code of both GPU and CPU of the Awkward Array.

The Awkward Array project is divided into 3 layers with 5 main components.



The work would involve automating tests for various “Kernel Functions” in the library. I would work on selecting appropriate input values for test cases of the kernel functions. Here Hypothesis Testing Tool would be used to let off the load a bit by using various statistical methods for testing.

The project has an automated testing set up through Azure pipelines. The work would involve building on this interface. Currently over 16,750 tests are generated and those are based on selection of inputs from arbitrary arrays. Work needs to be done to identify exceptional and extreme cases and conduct tests on those input values.

The Kernel Functions would soon be implemented in CUDA therefore this project work would prove to be the genesis of the understanding that the CUDA implementations are correct and good to work with.

Milestones

The milestones are quite flexible and cannot be estimated what types of problems might be faced while tackling them. Here the each goal can be stretched over a period of 2-3 weeks for over the duration of 3 months of the project (June to August, 2021)

1. Week 1

Getting familiar with awkward codebase - which includes going through the kernel functions and tests written in python and C.

Getting familiar with hypothesis testing libraries and suitable statistical testing methods that can be suitable for testing.

2. Week 2

Understanding extreme cases and noting down the type of functions that can be used for testing the different kernel functions.

3. Week 3-4

Experimenting with different statistical methods for getting the appropriate input values for each function.

4. Week 5

Understanding the validity constraints on the inputs and working on selecting the extreme cases and values that throw exceptions.

5. Week 6-7

Producing a mix of various tests that pass and test the major broad features of the kernel functions.

6. Week 8

Understanding and noting down the validity of various inputs and going deeper into understanding specific functions and their input constraints.

7. Week 9-10

Writing input test cases for checking exception throwing for specific functions in the kernel.

8. Week 11

This week would be kept as a buffer to allow testing and debugging the test cases and exception handling of the kernel functions.

9. Week 12

Documenting the tooling and tests used in hypothesis and changes made to the project. Preparing a presentation on the work done.