A lightning-fast stingray: Parallelizing stingray operations to analyze larger-than-memory datasets

OpenAstronomy
stingray

Mentors:

- Matteo Bachetti
- Daniela Huppenkothen

Student:

- Amogh Nirav Desai

Index

Final Project Report
OpenAstronomy - stingray

Introduction

This document precisely summarizes the work done during the tenure of the GSoC project as well as the future work needed for the thorough completion of the project according to the proposal.

stingray is a python library built to perform time series analysis and related tasks on astronomical light curves. The time taken and memory used for certain computations on large datasets in stingray is astronomical.

This GSoC project has located points of contention; optimized them to a certain degree and added support for analysis of larger than memory datasets. All the related code and changes can be found below.

Primary Work

1. Support for analysis of larger-than-memory datasets

a. **Description:** Due to the high memory consumption of the *AveragedCrossspectrum* and *AveragedPowerspectrum* classes, analysis of large datasets, ie. larger than 10,000,000(10Million) could not be performed. Using a python library Zarr, support was added for creating *AveragedCrossspectrum* and *AveragedPowerspectrum* from Lightcurves, EventLists and FITS files.
This has enabled analysis of large datasets up to 1,000,000,000(1 Billion) data points.

b. **Links:**
   i. Pull Request: #496

2. Benchmarks and profiling of stingray methods

a. **Description:** Served as the crucial step in the first phase of the project. All methods of stingray were profiled and benchmarked. A '*Benchmark*

Analysis Report' was created to discuss the results of the benchmarks performed.

b. **Links:**
   i. Benchmark Analysis Report - Benchmark Analysis Report 2020 - stingray
   ii. Pull Request for notebooks and other files - Benchmarking Stingray -> GSoC 2020 · Issue #477 · StingraySoftware/stingray

3. Integration of AirSpeed Velocity(asv)

a. **Description:** Integrated a tool for benchmarking stingray over its lifetime. Runtime, memory consumption and even custom-computed values may be tracked. The results are displayed in an interactive web frontend.

b. **Links:**
   i. GitHub repository: theand9/stingray-benchmarks: airspeed velocity(asv) benchmarks for an unladen stingray

4. Optimization of existing methods

a. **Description:** Various detailed, GitHub issues were opened based on the results obtained from benchmarking stingray. Pull Requests were created to solve these issues and optimize stingray.

b. **Links:**
   i. Issues: #481, #482, #483, #484, #485, #486, #487 #488, #489, #493
   ii. Pull Requests: #479, #490, #491, #494

Supporting Work

1. Update according to APE-17

a. **Description:** Updated the stingray package to the new Astropy infrastructure i.e. APE-17.

b. **Links:**
   i. Pull Request: #469
   ii. Documentation: APE_17 Update

2. Added documentation for Coding, Contribution, Testing and Community Guidelines

a. **Description:** Added coding, contribution, testing and community guidelines for consistency in the codebase and testing.

b. **Links:**
   i. Issues: #6
   ii. Pull Request: #474

Future Work

1. Proposed work

➢ **Parallelization:** The project has made stingray capable of handling larger than memory datasets, but the methods need to be parallelized to speed up computations and make full use of the newly added methods and Zarr.

➢ **Integrating numba**: Most parts of stingray are already optimized using numba. Integrating numba in all the necessary methods will also add speed improvements.

2. Other plans

- ➢ Solving the created issues.
- ➢ Optimizing stingray methods for less memory consumption.

Epilogue

Working with OpenAstronomy, stingray has been the most amazing experience for me. The summer has been anything but '*sunshine and rainbows*', but I wouldn't have asked for anything less.
Continuous learning and experimentation have helped me grow by leaps and bounds as a Computer Science student.

GSoC has also helped foster a love for Open Source within me. I am in love with *FOSS* and the idea of software that can be used by anyone in the world, developed by people all around the world, doing their bit to make the world a better place. It gives me a sense of fulfilment, I am committed to working on stingray and other Open Source projects in the future.

I would like to thank all the mentors from OpenAstronomyfor being very supportive, especially my mentor Mr Matteo Bachetti, without whose guidance this project would have not been possible.