Taylor Jacovich
PhD Student
Adviser: Dr. Alexander van der Horst
Department of Physics
George Washington University
Staughton Hall 301
707 22nd Street
Washington, DC 20057
tjacovich@gwu.edu
May 2, 2017

Design Specifications for a GRB Afterglow Distributed Computing Project

Part 1: Planning and Identifying Possible Issues

Introduction

Since the dawn of the *Swift* era, Gamma Ray Bursts have been recorded almost daily, with most of these bursts receiving follow-up observations to create lightcurves and spectra. These bursts have been repeatedly modeled using multiple methods, but no one has performed an exhaustive broadband modeling campaign on a large portion of the available data using a single model. With recent developments in modeling algorithms and software, now seems like an ideal time to undertake such a campaign. The amount of data available is well beyond the reach of a single, or even small team of researchers, but this sort of situation is perfect for enlisting the help and support of citizen-scientists.

This white paper outlines the preliminary ideas and design requirements of a distributed computing project based on the *boxfit* and potentially the *scalefit* program used to fit GRB afterglows using direct two-dimensional hydrodynamic simulations. The goals of this project would be to create an efficient and parallelized code that could be run on higher-end consumer hardware or workstations at the rate of fitting about 1 GRB afterglow per machine per day.

It would be advantageous to be able to automate data retrieval from archival sources for afterglows given the large quantity recorded during the *Swift* era and the massive amount of follow-up data provided in papers and in digital archives. To that end, we would adapt existing open data sources to work with the *boxfit* algorithm, with limited to no human intervention.

To spur adoption, as well as to support science-literacy among citizen-scientists and the population at large, We will produce a UI (User Interface) and accompanying UX (User Experience) that is both modern, efficient, and open. This UI would have to be recognizable and usable by a novice, and provide an accurate and informative explanation of the underlying science. This would be coupled with interactive ligthcurves and spectra, videos/podcasts, and explanations of the hydrodynamics among other things. Additionally, the program should allow visualization of the observational data, as well as the most recent intermittent lightcurve fit so that modelers can see what the program is outputting and make comments and alter the parameter space as they see fit with the ultimate goal of fostering a community of citizen-scientists in the Afterglow community.

Current Challenges

Currently, the biggest challenge is in moving the *boxfit* code to one that can be compiled for use on a CUDA enabled device. I have been toying with it in my spare time, but in order to make this feasible, I would have to have preliminary code running for testing by the Mozilla project deadline on

May 14th. I think I can produce this, as well as a basic website to house the code prior to submission, but I would likely be spending most of my time on that aspect of things.

Additionally, there is the issue of importing and converting data from multiple sources and in multiple formats, all potentially with varying units. These would need to be interpreted and converted as necessary. For archival data, this would be a simple job of identifying the archive format, but for researcher provided tables, this could be quite difficult. Creating a style guide for new data would be useful, as well as providing a place where researchers can submit data to the system, but it is a lot to expect of researchers to do all this specific work. In that instance a Machine Learning Algorithm may be able to help, as well as a potential web portal for users to help identify data formats once the community has been created.

As far as UI development, there are a lot of FLOSS(Free/Libre and Open Source Software) tools that can produce a modern and intuitive UI (I am writing this on the Gnome DE using LibreOffice as an example.) The big issue will be producing web and application layouts that are appealing and user friendly, but also convey the information they need and bolster the impact of the underlying science. Web design and initial promotional and communication products could be generated quickly, but a sustained and matured web and application presence would require at least an increase and expansion of my skill-set, but ultimately, it would be beneficial to have people with experience in these matters.

Part 2: Design Parameters and Constraints

Efficiency of the Code

There must be a noticeable speed-up in a CUDA or OpenCL enabled *boxfit* versus a single-threaded variant on reasonably modern hardware. A background process with no more than 30% utilization of the overall system should be able to perform a proper fit within a matter of a few hours for this to be tenable as a distributed computer system, assuming the average power on time of a home system is around 8 hours a day. The current single threaded version puts a 100% load on a single core for several minutes to perform one off axis iteration of lightcurve or spectra production. Timing of a chi-squared fit must be performed.

Multiple frequencies should be able to be reasonably fitted at once to properly fit the data. Ideally at least two frequencies should be able to be fitted so that samples can be taken both above and below the cooling break. This will require a slight modification of the code, but I do not think it will create a significant increase in the required calculations as *boxfit* seems to generate the emission of each boxfile, and the number loaded can likely be optimized and shared memory can be used to minimize overlap. The name of the game is parallelization and memory-management for this one.

Requirements of the Distributed and Server-Side Systems

Initially, these systems will likely be based off the BOINC API as this is a robust and well implemented distributed computing network that hosts both SETI@home and FOLD-It@home to name a couple of prominent examples. The documentation for implementation of CUDA and OpenCL accelerated code is well-written and relatively simple to implement. Setting up a server to handle data requests and ingest incoming fits is also well documented. The problem with the BOINC system is that none of the applications are really set up for user intervention. I will not discuss the details of that here, as those details are best saved for the discussion of the UX, but suffice it to say that we will likely either need to build on the BOINC interface, or create our own as we go. These are details that will need to be ironed out once the initial implementation is produced and we can see how the data needs to

flow and what loads are placed on the control servers. Amazon web hosting may be a wise way to begin with this.

The User Experience (UX)

The User experience must meet the following criterion:

a) It must be sufficiently intuitive and robust that even a novice can navigate the basic functionality.
b) It must be modern and efficient such that it would not look out of place on any current operating system. To that end, it must also be platform agnostic and open.
c) It must be sufficiently light that it can run on most consumer hardware, even if the science objectives cannot be reasonably achieved on that hardware.
d) The web component must be equally lightweight and browser agnostic.
e) Any web page or application must contain detailed information on data acquisition and processing, as well as how different wavelengths of light are treated.
f) There must be sufficient user interaction with visualizations of the data and the science such that the user can feel immersed in the activity and grasp the physics through hands-on activities
g) There should educational content in several media forms including animations, videos, and audio podcasts that provide context and continued information on the state of GRB physics
h) Branding must be unobtrusive and memorable.

A UI similar to the one implemented for Zooniverse could be helpful, or one similar to a video game. A Steam-based distribution could potentially let this become something akin to universe sandbox, except with a much more direct citizen-science bend. Again, we will have a better understanding of the needs of this UI as we implement the initial code, and begin deciding the direction to build for the community.

Ingesting Data from External Sources

Astronomical data from orbital observatories tends to be well pipelined and organized in a methodical way. Likewise, most observations from larger observatories also tends to get the same sort of treatment. These will be relatively easy to ingest and convert into data that can feed the computing nodes. The challenge will be in ingesting data from more irregular sources, like from tables provided by individual researchers or from sources like the AAVSO. The variety in data reduction and preperation will need to be dealt with by hand initially, but scripts will quickly be developed to minimize ingest time, and hopefully the process can be fully automated for most sources, and a machine learning algorithm trained for the remainder.

Ultimately, we would like to produce a universal data template with the relevant metadata, potentially in the same vein as the .fits file(It could even be the .fits file if we can manipulate it properly) used in optical observing that contains large quantities of observing data for a given object, and provide a methodology for saving data that would make it straightforward to ingest data into these sorts of projects.

Tentative Timeline for the Project

May 14, 2017	Have CUDA code operational and preliminary time data and website set.
June 2017	Demonstrate initial BOINC implementation, along with at least one complete broadband fit Have a functioning website with an initial releases of supplementary content Begin work on ingest code
July 2017	Have rudimentary ingest code that can reasonably feed well formatted data Demonstrate BOINC proof of concept on limited network of local computers Have relatively stable content release cycle Optimize and improve CUDA code to fulfill design requirements
September/October 2017	Content release cycle finalized Continued work on website and content Have several afterglows that have been modeled by the networked computing nodes Continued work on ingest code Optimize and improve CUDA code to fulfill design requirements Give the network a snappy name and branding
December 2017	Take the BOINC network fully online (This may be shifted depending on state of ingest) Have website content that reflects full realization of the BOINC system or discussion of preliminary results of the system. Begin laying out design for desktop application Begin laying out ways for community to contribute to ingest Discuss ways to market the new system in cost effective (read: free) ways. Optimize and improve CUDA code to fulfill design requirements Begin OpenCL port
2018	Optimize and improve CUDA code to fulfill design requirements Take BOINC network fully online Begin alpha tests of desktop UI for beyond BOINC capabilities Community engagement increases and sustainable growth continue marketing and promotional activities Produce fully interactive content for citizen-scientists to explore various aspects of GRBs and the evolution of the afterglow maintain multimedia content stream Complete OpenCL port
2019/2020	Optimize and improve code to fulfill and surpass design requirements, improve on original *boxfit* and *scalefit code* wherever possible. Beta and full release of desktop application finished ingest algorithm with community backup system create forums and other spaces where users can interact and contribute bug report system continue with website content updates and improvements provide statistics and pre-prints of any papers stemming from the program.
2021+	sustained output, update, and operation of the project potential for expansion as the field grows and evolves potential for similar open projects like it

Final Comments

I will readily admit that I am not an expert in a lot of the items on this list, but I am passionate about astrophysics and am excited to have a project that has the potential to reach so many people who may be interested in science. In periods like what we are currently experiencing, now is the ideal time to create systems like the one described above, where anyone with a desire to contribute to meaningful research has the tools and support to do so. These methods may be adapted by use on other aspects of physics/astrophysics or even in other scientific disciplines once they are implemented.

The project I have outlined above is ambitious, it requires not just exacting detail in the science, but also in marketing the approach and project to the general public. It will be not just an exercise not just in producing efficient and error free code, but in creating an ecosystem of incoming and outgoing observational and theoretical data packaged elegantly and transparently for presentation to other researchers and the general public. That is incredibly daunting, but also really exciting. It allows for tackling astrophysical problems not just in a direct and scientific sense, but also for making the general public an active participant in cutting edge science.