Google Summer of Code

GSoC 2022 proposal

Konstantinos Petridis

NetworkX

April 11, 2022

Abstract

This is a detailed review of the experience, education and set of skills that I have cultivated throughout my academic journey. My excitement about contributing to such an extraordinary project and my confidence about possessing the necessary scientific knowledge, generate strong motivation and desire to participate in GSoC and the creation of amazing, open-source technologies.

1 Personal Details

1.1 Contact information

[redacted]
GitHub: https://github.com/kpetridis24
GitLab: https://gitlab.com/kpetridis
Linked-in: https://www.linkedin.com/in/kpetridis/
[redacted]

1.2 Education

Integrated Master in **Electrical & Computer Engineering** at the **Aristotle University of Thessaloniki**. Currently on my 5^{th} year of studies with a **Major in Electronics & Computer Science**.

1.3 Skills

Technical skills	
Programming	C, C++, Python, MATLAB, Java
Libraries/API	FreeRTOS, CUDA, BLAS, OpenMP, MPI, Cilk, p-Threads, NumPy, Pandas, Scikit-Learn, NetworkX
MCU	ATSAM, Arduino
Tools	Git, Docker
DBMS	MySQL, MongoDB
Others	Parallel Programming, Distributed Programming, Object Oriented Programming, Graph Theory, Dynamic Programming, Machine Learning

1.4 Experience

Software Engineer as a member of the **On-Board Computer** team in ***SpaceDot***, which is an open-source, student team. Our current project includes the construction of ***AcubeSAT***, a nano-satellite, scheduled to be launched into space in 2023.

Contribution

Development of the on-board software, responsible for management of data, flowing throughout the satellite's sub-systems. Major functionalities include message parsing from the ground station, telecommand request processing, telemetry creation, parameter sampling, storing and housekeeping, statistics reporting, error handling and logging.

Learned

Low level embedded programming of micro-controllers
Creation of sufficient and descriptive documentation
Modern C++ skills
Unit testing
Code reviewing
FreeRTOS

1.5 Commitment to GSoC

My obligations during the period of GSoC are basically the University exams. However, they will be over about a month before the first submission of the project, based on the existing timeline. Apart from that, I am able to handle it even for the small period of time in the beginning of GSoC and I have the intention of total commitment to the project. My plan is to work as consistently as possible, making my contribution to the project a **top priority** throughout the entire timeline.

2 About the Project

NetworkX currently provides a **Graph Isomorphism** implementation, which uses a recursive version of the **VF2** algorithm. However, **this paper** describes an improved version of the original method, called **VF2++**, which we are planning to implement. Attempts have already been made to implement the main algorithm, but the goal of the project is the full implementation of the algorithm for all problem and graph types in an iterative manner. Another useful paper that discusses **VF2++** is **this one**.

What we are trying to achieve, after verifying that both the input graphs have the same number of nodes and the exact same degree sequences, is to compute a **mapping**, such that every node from a graph A, is matched to a node of graph B. If the mapping can be extended such that, every node from the first graph is matched to a node of the second graph, then the two graphs are considered to be Isomorphic.

According to **VF2** the extension of the mapping in every iteration creates a tree, in which every node of the tree, is a candidate pair of nodes, one from the first graph and one from the second graph, that is currently being examined. If the two nodes can be matched, the mapping is extended and the algorithm continues accessing deeper nodes of the tree. If the matching cannot be established between the examined pair, the algorithm goes up on the tree, in previous nodes and searches for, new matching pairs to extend the mapping.

The goal of **VF2++** is to find a matching order in which the algorithm is able to recognize inconsistency or prune the infeasible branches on the highest levels and goes deep only if it is needed. This **pre-process** should boost the performance and the efficiency of the original algorithm, which is what the existing benchmarking analysis indicates.

3 Project Timeline

TABLE 1 Timeline for the implementation of VF2++

April 4 – May 20	**Personal Research** During the period starting from my proposal, until the announcement of the accepted GSoC contributors, i will do extensive research on the topic of **Graph Isomorphism** and specifically the **VF2++** algorithm. The goal is to have established a clear perception and understanding of the project-related concepts, before the actual implementations begin.

May 20 – June 12

Community Bonding

During this period, i keep close contact with the mentors, discussing my thoughts on the strategic approach to complete the project, the actions that should be taken, and the sub-problems into which we shall decompose the initial, more complex problem. Simultaneously, i become more and more familiar with both the software of **NetworkX** and its community, by exploring the different functionalities provided by the library and resolving each possible question that will emerge.

June 13 – July 25

Development Phase 1

At this point, we should have managed to lay the foundations of our projects, clarify the most significant concepts and have a comprehensible view of the desired outcome. Until the end of this period, the **VF2++** algorithm should have been implemented, at least at its preliminary, functional form.

July 25 – July 29

Review Phase 1

The first **pull request** is created and the review of the first assignment begins. Feedback about possible optimizations is provided by the community and the mentors.

July 29 – Aug 20

Development Phase 2

In this phase, suggested optimizations are applied, possible extra features are added and any issues of the so-far progress are resolved. After this process, the final product should have been completed and incorporated into the software of **NetworkX**, which may require some additional modifications.

Aug 20 – Sept 1

Review Phase 2

Final assessments of the final result. Hopefully at this point there are no major bugs or inefficiencies and the algorithm is sufficiently tested and extensively reviewed. The existence of **documentation**, describing the implementation in detail, is of undisputed importance and should be perfected. Finally, the creation of **Unit Tests** to ensure the validity of the results could be a quite assisting add-on. This phase could also include discussions about further enhancement of the implementation, outside the scope of GSoC.

4 Motivation

4.1 Why GSoC?

First and foremost, I am a huge supporter of Open Source and I believe that it gives infinite opportunities to developers and engineers to expand their knowledge, acquire new skills, explore new fields and contribute actively to the evolution of scientific domains. Many passionate people collaborate to create awesome new technologies and make amazing software available to the scientific, research and academic community.
In addition, it's a unique opportunity to interact with highly skilled and knowledgeable individuals, and work as a team towards a significant goal. The journey is for sure going to have a tremendously positive impact on both my skills and experience and is going to open a new chapter for me being an active Open Source developer.
Lastly, I am willing to take advantage of this summer period, since I will be available and free of any obligations, to do what I love, challenge myself to learn, grow and face new difficulties, adapt to a not-so-familiar environment and become better at what I do.

4.2 My interest in NetworkX

Currently i am starting to become an active **NetworkX** contributor, since i am working more and more on issues, and proposing new features, **such as this one**. I have already created three **Pull Requests** (**1**, **2**, **3**) and i am currently developing a DTS (Dynamic Topological Sorting) implementation for **NetworkX**. The related issue **is located here**. After this, i plan on proposing my version of **Community Detection** for **Overlapping Communities**, which i have completed and is currently being tested, as well as discussing the parallelization of already existing implementations.

4.3 Why this project?

One of my most loved fields of interest is **Graph Theory**. I have already put significant effort and time on my graph-related projects, which can be found on GitHub and include **Triangle counting in large sparse graphs**, **k-NN**, **Community Detection**, as well as others. Particularly, **Graph Isomorphism**, apart from being a super exciting and interesting problem to solve, is also suitable for both my experience and programming skills, for which i am highly confident. In addition, i have used the **Networkx** library in a lot of my recent projects, which makes me familiar with the tools that it provides.

4.4 Why choose me?

During my academic journey, I have developed the skill to be able to manage extreme amounts of workload, work efficiently under pressure, meet deadlines and do targeted research to enhance my understanding of what I am dealing with. One of my most prominent characteristics when I work towards some goal, is the never ending persistence and effort I put into it. I am confident about my fast-learning skills and the organized, structured approach that I take to solve every problem.

Also, this is not a short term goal for me. I will actively try to be part of Open Source communities in the years to come, and devote a lot of time writing code, contributing and presenting ideas on topics of my interest. I think you should definitely give me a chance to prove my capabilities to you and the fact that, I have what it takes to fulfil my mission for the summer.