Google Summer of Code 2019 Proposal

Graph compression on the development history of software

Thibault Allançon
haltode@gmail.com

Mentor: Stefano Zacchiroli

1 Objective

Implement graph compression techniques for the main data model (Merkel DAG) to make it fit in memory.

2 Rationale

This ever-growing graph containing billions of nodes and edges is at the center of Software Heritage mission to archive the entire software commons. Enabling in-memory processing of the graph will greatly enhance internal queries over the whole dataset.

3 Approach

Research time: from April 9 (end of application period) until May 27 (start of coding period).

- Get familiar with Software Heritage infrastructure and data model. *(1 week)*
- Define desired graph operations and characteristics.
- Synthesize graph compression papers and approaches. *(2 weeks)*
- Evaluate feasibility and compression rate of different framework/techniques. *(2 weeks)*
- Set future implementation goals.

Coding time: from May 27 until August 26 (end of coding period).
Implementation is still undefined as it might involve starting from scratch or building an API around an existing framework.

- Compression code (from naive edge lists to compressed format). *(3 weeks)*
- In-memory loading of the compressed graph. *(2 weeks)*
- Graph operations and query code. *(3 weeks)*
- Link code with the rest of the infrastructure. *(2 weeks)*
- Set up automated unit and integration tests.
- Write technical and architectural documentation. *(1 week)*

This subject is prone to a lot of experiments, meaning it is hard to define a strict plan and timeline in advance. The one proposed above will certainly vary during the internship.

4 About me

I will be in South Korea until the end of June before coming back to France. The timezone difference should not be a problem since you can always contact me with IRC or email. Furthermore, I plan to write regular recaps on my personal website [1] and to always keep my mentor up to date.

[1] https://haltode.fr/gsoc2019.html

5 FOSS contributions

My main FOSS contributions consist of personal and school projects, hosted on GitHub:

- Small experimental x86 operating system in C.
- Re-implementation of Git version control in Rust.
- Implementation of machine learning algorithms using Matlab.
- Emulation of an entire computer (hardware, operating system and compiler).
- Minimal Lisp-like programming language.
- First year school project: Civilization-like multiplayer game made using Unity.
- Second year school project: optical character recognition software written in C.

I am an active member of the French association Prologin [2], which goal is to introduce young students to the world of programming and algorithms. Concerning open-source projects, I worked on the contest finals environment:

- Creation of Prologin 2018 finals game in C++.
- Contributions to the client-server AI match maker used for the contest.

Here are the differentials I worked on to start contributing to Software Heritage: `D1217`, `D1235`, `D1226`, `D1242`. As a more far-reaching contribution, I would also like to work on the Rust crate loader.

[2] `https://prologin.org/`