Project Software Engineering Proposal

Volunteer Computing in the Browser Using WebAssembly

Erik Kooistra
May 29, 2019

1 Introduction

Distributed computing is extremely important to modern research; it allows scientists to perform extremely large computations in a reasonably short time. For some research, however, a large distributed computing grid is not available; sometimes researchers have to turn to so-called volunteer computing. In volunteer computing, users voluntarily make their computing resources available to scientists through programmes like Folding@Home or GIMPS. One issue with such projects is that they often require the user to install something on their computer, and that may be a barrier for some. In this project, you will be developing a service for volunteer computing in the browser. Your framework should allow researchers to upload C code written with a specific framework that you will compile to webassembly/JS, and users should be able to select a research project to have their browser work on. The service should be user-friendly and have some sort of reputation system to filter out users who wish to sabotage the system by submitting incorrect results.

2 Objective

The final product should be a website where users can select projects to perform computation for and a platform where researchers can upload there projects that will get compiled an distributed to the runners on the users.

3 Requirements

We present the following requirements for several different aspects of the project. These requirements are intended to sketch an outline of the intended product and may change during the project.

- The product shall have a C library where researchers can write their compute kernels in
- The product shall have a back-end where researchers can select the trust level and input space.
- The product shall allow users to score points for the work that they have performed.
- The product shall have a reputation score based on amount of correct results submitted

- The product shall submit the same job to multiple users to perform verification.
- The product shall allow the user to select how many workers he want to spawn on his client.
- The product shall fingerprint the client to detect if mutiple users are the same.

4 Project predictions

The part of compiling C to webassembly and running this in the browsers has good documentation. The difficult part is the distribution of the jobs and keeping track of who is working on what, and if the client is still alive. And keeping it running inside the browser for long periods of time. Implementing the detection of mallicous workers can also by quite tricky. However implementing a some simple heuristics should be enough as a proof of concept, besides having multiple clients doing the same work.