university of groningen

faculty of science and engineering

Highly Distributed In-Browser Computing

Bachelor's Thesis

July 2019

Authors
George Argirousis
Tolga Parlan

Primary Supervisor
Who?

Secondary Supervisor
Frank Blaauw

CONTENTS

ABSTRACT

Here be the abstract

1 INTRODUCTION

Modern web relies increasingly on Javascript, which is used almost ubiquitously in client-side of any website and even increasingly on the server-side through frameworks such as Express. Client-side Javascript often has the purpose of making websites more reactive to the user actions by offloading some application logic to the client's browser.

As it is becoming more and more common to send tasks to the client-side to reduce the communication overhead, Javascript and related web technologies are undergoing rapid innovation in terms of performance and quality. [TRY TO FIND SOMETHING ON JS OPEN SOURCE ENVIRONMENT]. This has been exposing a considerable amount of computing power to the website owners since presumably everyone visiting their page run code written by them, on their browser.

[SUPER COMPUTER COMPARISONS]

Distributing a large pool of computing tasks to many computers across internet and let them run the calculations is an idea which now has more than two decades of examples, including successful project such as BOINC [REFERENCE]. Such projects have been usually termed **Voluntary Computing** since the participants would actively volunteer to run such tasks on their machines, usually via signing up on a website and downloading a purpose-made program [BBHOFVC paper]. [TALK ABOUT BROWSER BASED VERSION OF THIS]

Another idea in this field which comes from a similar premise is **Gray Computing**. Gray computing refers to a non-voluntary category of task distribution where users don't explicitly consent to running the given computations on their browsers, but the implicit consent which comes from the fact that any website can run arbitrary Javascript on it's visitors computers, is used to create a similar architecture. This environment presents different technical challenges and opportunities, which are explored in the next chapter. [CITE THE PAPER]

The lion's share of revenue for most websites that do not explicitly sell products come from advertisement [FIND STATS]. Ads can present a series of complications to internet users, [MAKE A LIST, FIND STG ABOUT THIS]. The negative user reaction against ads is tangible in ever growing use of ad-blockers [FIND STATS]. [RESEARCH IF ADS DECREASE USER ENGAGEMENT]

This project asks the question whether the aforementioned processing power can be harnessed effectively to create a serious alternative revenue stream for many websites, nullifying or decreasing the need for ads. To test the feasibility of the idea of webmasters using gray computing practices to generate revenue, we have implemented an npm package [EXPLAIN] which integrates seamlessly with an Express server, has an easy to use interface for programmers, and can keep a low memory overhead while distributing large tasks among the website visitors. We intend this to be an open-source project to which anyone can contribute with new features or performance improvements, or simply fork from and use as a starting point in their own projects. We believe that the rigid architectural choices such as focusing on a limited set of algorithms, or [??] have in the past harmed the usability of similar projects in practical environments. This topic is discussed further in the next chapter.

This paper is structured as follows: In Chapter 2 we give a summary of related work, explaining how we have benefited from the design and technology choices of similar projects to ours. The chapter further quotes from survey papers which have examined many past projects, and goes on to explain in which ways we have attempted to differ in our project in order to produce a solution that can be adopted easily by people for diverse tasks. In Chapter 3 we summarise our code architecture, and talk about the main components of our program in more detail. Chapter 4 focuses on evaluating different types of tasks and their performance on our system. Finally in Chapter 5 we present our conclusions and in Chapter 6 we present ideas that can be considered to improve the project in the future work.

2

RELATED WORK

3 ARCHITECTURE

4 EVALUATION

5 CONCLUSION

6 FUTURE WORK

BIBLIOGRAPHY