Browser Based Distributed Computing
TJHSST Senior Research Project
Computer Systems Lab 2009-2010

Siggi Simonarson

October 28, 2009

Abstract

Distributed computing exists to spread computationally intensive problems over a wide array of machines in order to obtain results more quickly than possible on one machine. This approach requires a large number of less powerful machines in place of one more powerful, more expensive, supercomputer necessary with a traditional approach. The largest array of such computers that exists today is a decentralized network of machines that communicate with one another over HTTP through web browsers known as the Internet. This research project seeks to combine these two ideas by harnessing the power of the Internet through widely available HTML and Javascript in browsers with PHP on the side of the server to perform large problems with relative ease. In addition to the framework, an easy to use web interface that would allow future researchers to set up volunteer networks with little knowledge of web technologies will be implemented for accessibility.

Keywords: Distributed System, Parallel Computing, Volunteer Computing, Browser Based

1 Introduction and Background

1.1 Problem

Modern attempts at Internet based volunteer networks all require the user to download and install some software, either a standalone program, or the Java Runtime Environment to contribute their processor cycles to the cause they wish to support. This project seeks to answer the question, can a framework be created that is more user friendly for both the volunteers, and the organizers of volunteer computing networks, using current web technology that require no installation or technical knowhow on the part of the volunteers.

1.2 Introduction

This project aims to create a framework that allows researchers to harness the tremendous number of nodes available over the Internet for volunteer computing. A web interface for project management will allow for a manager-worker task distribution model to be set up with relatively little knowledge of web technologies on any operating system. If the framework comes to fruition and is satisfactorily accessible, fault tolerant, secure, and efficient, a sample problem to demonstrate the use of the framework will be implemented. If possible, that data will then be sent to one of the various volunteer computing projects in existence today.

1.3 Previous research

Several research projects have been done in the field of volunteer computing. A project based out of MIT seeks to provide a similar framework to developers using Java instead of Javascript, which they hope to increase speed, but severely limits the number of nodes they can access. With the current increase in the speed of Javascript in modern browsers due to the advent of web based applications, Javascript will the fastest, most accessible language available. A background in web development is essential to the development of this project. The project will be coded almost entirely in PHP and Javascript using HTML and CSS to display the interface and AJAX for message passing.

2 Procedure

To begin, a cursory working model of the manager worker interactions between the server and the browser will be established to assess the validity of the idea. Once a rough model is established, the storage scheme will be implemented to keep track of results, statistics and the data that needs to be calculated. From there a focus will be placed on making the code as general as possible to convert it from a model to a framework that can be used for any number of applications. A focus will be placed on a way for those implementing the framework to easily adapt it to their needs, fault-tolerance, security, scalability, and interface. From there, the framework will be released and the project will be working towards a implementing a useful distributed computing project that will display the features and test the validity of the framework.

3 Results

3.1 Current State

At the current state, a user visits the contributor website and receives two numbers from the initialization function, and a folder is created on the server for the new user, user values are incremented, and files are created to accommodate the new user. When the user selects the Start button, their Javascript engine takes over and adds the two numbers together, and using AJAX it sends the result to the server. The server takes this result, with the data pair that was originally sent, and adds them to a results file. The server then increments the user value, and the data pair value and sends the user a new data set. The user then requests a new data pair, receives it from the server, calculates the new pair in Javascript, and sends the result back. This process repeats until the user quits.

In its current state, the administration section is not as robust as it will eventually be. The only two options are to view results and purge data. The view results function takes the data file in which all the results have been calculated, and displays it in a web friendly way that the administrator of the project can easily view. The purge data function came out of necessity. Having to always go back and reset all user values, id values, and results data became rather tedious, so the process was automated to the largest degree

possible. User folders still have to be removed manually.

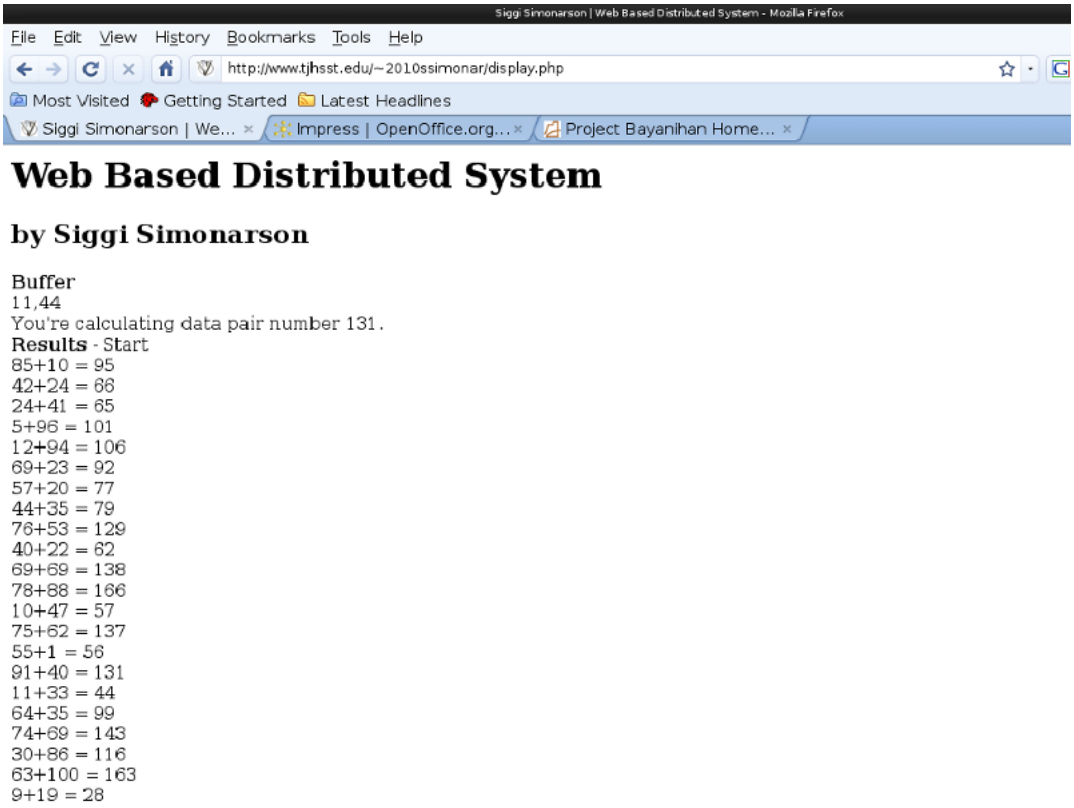

Figure 1: Screenshot of current state.

3.2 Ideal State

Ideally at the end of this research project, the simple addition calculation will be replaced with the ability for the administrator to define a calculation to perform. The data set will also be different, and suited specifically to the project being run using the framework. The contributor interface will be much more user friendly, with the ability to set the speed (limiting processing cycles), stop the calculations, and view more comprehensive statistics of their calculations. The administration section will also be far more robust than the current state. As previously stated, the administrator will be able not only view the results, but add new data, and change the calculations performed

on the data. The administrator will also be able to see more comprehensive statistics as to the state of their project.

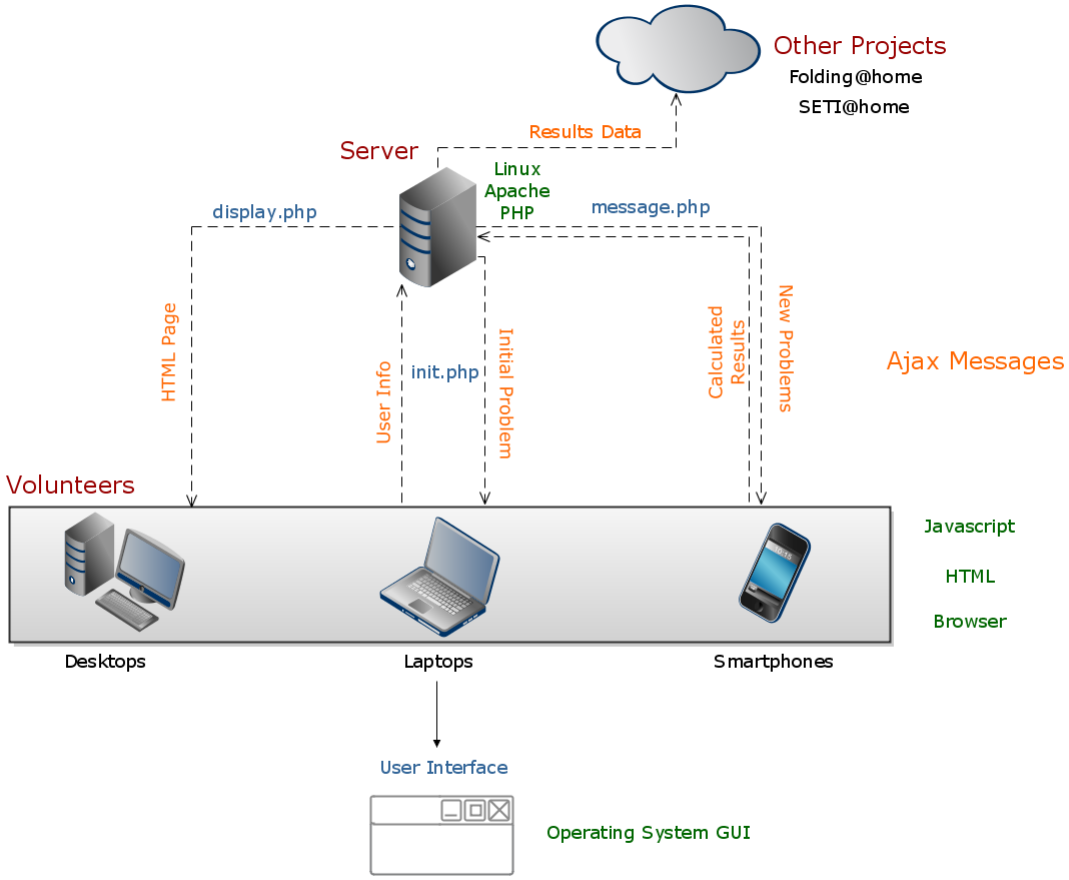

Figure 2: Overview of project.

4 Discussion

The results will be based on the completeness of the framework, judging based on its adaptability, fault-tolerance, security, scalability, and interface. The successfulness of the project, and the answer to the question being researched

relies heavily on the speed of Javascript. If a significant speed improvement can be seen with an increase in volunteers, the project will be successful. Even with a slight speed improvement, this will be magnified as Javascript engines in more modern browsers like Google Chrome become more efficient and powerful.

References

[1] Luis F. G. Sarmenta, "Sabotage-tolerance mechanisms for volunteer computing systems", *Future Generation Computer Systems*, 2002.

[2] L. F. G. Saramenta and S. Hirano, "Bayanihan: Building and Studying Web-Based Volunteer Computing Systems Using Java", *Elsevier Preprint*, 1998.