1 Motivation and Background

The scientific progress in the past few decades has been phenomenal. Emerging fields such as bioinformatics and computational biology have provided us with deep insight into how the human body works. Companies such as Illumina are now able to sequence the whole human genome for less than $1000.

Most people use the Internet for simple tasks such as reading the news or browsing Facebook. Because most people have very powerful computers, there is a lot of unused computing power.

What if this untapped computing power could be used to solve the research problems? Atlas is a distributed computing system that allows for collaboration over Internet browsers. It leverages this wasted processing power to help leading researchers and companies compute difficult tasks. The platform aims at maintaining similar speeds to available cloud computing services while running these tasks, while doing so at a much cheaper cost.

2 Overview

Atlas is a platform that provides a grid computing network over the internet to connect researchers to unused computing cycles. The platform will allow for the upload of compatible
Javascript code. As an alternative, users can upload code in a language of their choice, and a computer-assisted translation pipeline will cover the code into a useable format.

After code is uploaded, Atlas will decide how many browsers it must connect to in order to run the research problem in an efficient manner. It breaks the entire research problem into small chunks of code, and then calculates a plan to distribute those chunks for optimal computation.

To actually solve the research problems, Atlas will provide a single line of JavaScript code that Web site owners can embed. When a person visits the Web site in question, the platform’s deployment features will open a stream between the web server and the client. The manageable chunks of research problems will be sent to the Web site, computation will occur in the browser, and the results will be sent back to the server. To incentivize Web site owners to use the Atlas platform, a small fee will be payed for each computation gained from their web traffic.

Finally, the results from each browser will be combined to form a complete solution to the initial problem, and the researcher will be informed that the solution is ready.

3 Technologies Employed

To develop the initial version of Atlas, the following technologies will be used:

• **Go Web Server**: Built in support for concurrency, and has been tested to work well with distributed networking.

• **WebCL**: Technology that allows for computation to exist in parallel between GPUs and CPUs through the browser. This will allow Atlas to quickly compute research problems on machines.

• **JavaScript**: This allows research problems to be sent to the Web browser, and allows the platform to easily be deployed.
• Jasmine: The testing framework for the application.

4 Competition and Advantages of Atlas

Several other players already exist in the same space, mainly cloud computing platforms like Amazon Web Services, Digital Ocean, and Microsoft Azure. Furthermore, many universities with leading researchers already own infrastructure for and deploy their own cloud computing platforms.

At the start, Atlas aims to compete with mainstream cloud computing platforms. Several key competitive advantages exist, which make Atlas a viable contender in the market. Specifically:

1. Ease of Integration: The only action researchers must take to get onboard is to upload their code securely to the platform. The only action Web site owners must take to start earning revenue is to include a one-line deployment script.

2. Code Translation: The platform is the only one of its kind that will employ WebCL technology for native CPU/GPU access. This allows for code to run significantly faster.

3. Resource Access: Development of Atlas is taking place at Yale University, which is the center of a thriving research environment.

5 Deliverables

Work on Atlas will be split over two semesters. Deliverables will be split as follows:

Fall 2015

1. Construct an initial version of the Atlas server and delivery system.

2. Explore WebCL technology, and evaluate it’s performance for this application.
3. Run the built platform on a set of 100-200 browsers, and measure the performance against existing cloud computing services, like Amazon Web Services.

4. A brief 8-page paper summarizing *Atlas* and the experiments and results obtained over the semester.

*Spring 2015*

1. Build interface for automating code upload and translation into a format that *Atlas* can use (i.e. allow a user to upload non-JavaScript code, and still have it be useable).

2. Automate the process of splitting up a script into small chunks to distribute to Web browsers.

3. Publish a research paper on the findings.