University of Victoria

CENG/ELEC/SENG 499 - Spring 2014

Improvement for SHOAL: A Dynamic Web Cache Publishing Tool

Progress Report #2

March 7th , 2014

Project Number: #09
Project Title: Improvement of SHOAL - A Dynamic Web Cache Publishing Tool
Faculty Supervisor: Dr. Daniel German
External Supervisors: Dr. Ronald J. Desmarais, Mr. Michael Patterson
Team Contact: 499shoal@gmail.com
Personnel:

Name	Email	Student No.
Y Nguyen	nguyeny@uvic.ca	V00705206
Mike Chester	mchester@uvic.ca	V00711672
Erik Afable	eafable@uvic.ca	V00692209
Hao Lu	therealhaolu@gmail.com	V00724916
Anita Katahoire	anitakat@uvic.ca	V00620051

1.0 Progress to Date

1.1 Framework Conversion

1.1.1 Selecting the Appropriate Replacement

The underlying framework that powers Shoal, Web.py, is sufficient for Shoal's current use case. However, in order for Shoal to support larger workload and supports more concurrent clients, Web.py proves to be too simplistic and lacks scalability. A number of other existing Python frameworks have been considered as a replacement for Web.py

Django: Django is a powerful web framework that provides full end to end support for the system. However, Shoal does not require many of the tools offered by Django. The framework proves to be too bloated for our domain.

CherryPy: CherryPy is a lightweight and efficient framework. However, it does not offer any advantage over the current tool, Web.py, and therefore, does not justify the switching of framework.

Tornado: The Tornado web server was chosen due to its high scalability and more sophisticated support for building asynchronous web servers . The team members are also more familiar with the technology and have previous experience working with it.

1.1.2 Progress of the Conversion From Web.py to Tornado

Web.py and Tornado, though of similar principles, differ quite significantly in terms of implementation. So far, our team has accomplished the following:

- Transitioned the majority of the code for Shoal Server to the Tornado framework.
- Integrated the RabbitMQ worker into the Tornado IOloop, as well as generated the handlers to service requests.
- Update the configuration settings so they can be more easily managed and updated.
- Created a periodic callback within the IO loop to delete inactive squids. Effectively we have integrated the 3 threads of Shoal into the main Tornado IOLoop. We just need to iron out a few details before we take steps to analyze the performance difference between the two versions.

1.2 Experimenting with WebSockets

Some experimentation was done with WebSockets to allow dynamic updates to the web GUI interface, that allowed it to be updated everytime a message was received from RabbitMQ. Although the websocket experimentation was some what successful, we decided to scale back on the implementation for now and finish up on the outstanding milestones. We may revisit adding WebSockets to create a more dynamic interface but for the time being we deemed them a lower priority.

1.3 Web User Interface Improvement

1.3.1 Business Website

The business website for the Shoal by 499 project has been updated to include a web front and deliverables. It has been built on the Bootstrap framework which has been included in the repository.

1.3.2 D3 + Google Maps Implementation

A template GUI has been created in D3 using the google.maps.Map object. Currently, the station information (i.e. the latitude and longitude points) are collected into a station.json object and the UI loads the JSON file to update the geographical locations on the map. The JSON file currently uses only static data.

2.0 Revised Milestones

ID	Task Name	Start Date	End Date	Duration	Assigned To	Percent Complet e
1	Replace Web.py with the Tornado web server.	2014-01-28	2014-03-18	1 Month, 3 Weeks	M. Chester Y Nguyen	90%
2	Implement a new algorithm	2014-01-28	2014-03-18	1 Month,	M. Chester,	0%

	to speed up computationally expensive operations.			3 Weeks	Y. Nguyen	
3	Improve the web GUI	2014-02-04	2014-02-18	1 Month, 2 Weeks	E. Afable, H. Lu, A. Katahoire	40%
4	Idea Pitch	2014-01-27	2014-01-28	2 Days	E. Afable, M. Chester, A. Katahoire, H. Lu, Y. Nguyen	100%
5	Progress Report #1	2014-01-28	2014-01-31	4 Days	E. Afable, H. Lu, M. Chester, Y. Nguyen, A. Katahoire	100%
6	Progress Presentation preparation	2014-02-18	2014-02-21	4 Days	E. Afable, H. Lu, M. Chester, Y. Nguyen, A. Katahoire	100%
7	Progress Report #2	2014-03-04	2014-03-07	4 Days	E. Afable, H. Lu, M. Chester, Y. Nguyen, A. Katahoire	100%
8	Public Demo Preparation	2014-03-20	2014-03-27	1 Week, 1 Day	E. Afable, H. Lu, M. Chester, Y. Nguyen, A. Katahoire	20%
9	Final Report	2014-03-25	2014-04-04	2 Weeks	E. Afable, H. Lu, M. Chester, Y. Nguyen, A. Katahoire	0%

3.0 Remaining Tasks

1. Finish framework conversion:
    a. Address comments from supervisors regarding current pull requests.
    b. Clean up code (remove unused code and files).
    c. Integration testing of system with Tornado.
    d. Update documentation.

2. Finish GUI improvements:
    a. Read and correctly parse the JSON file produced by the shoal server to obtain squid server coordinates.
    b. Display squid servers properly on the map from the above coordinates.
    c. Use Ajax to reload the website each time the above mentioned JSON file is updated.
    d. Animate the map to show the changing loads of each squid server and the changing geographic points.

3. Finish project and report and preparation for public project demo.