Improvement of SHOAL

A Dynamic Web Cache Publishing Tool

A. Katahoire E. Afable M. Chester H. Lu Y. Nguyen

Breakdown of Tasks

Enhancement Team (ET):

- Improving the GUI
- Analyze the old UI and gather new requirements

Core Application Team (CAT):

- Framework conversion
- Algorithm improvement

SHOAL Milestones

1. Replace Web.py with a better web framework.
2. Implement a new algorithm to speed up computationally expensive operations.
3. Improve the web GUI.

Choosing our Web Framework

Considered:

- Django: Too bloated for our project.
- CherryPy: Doesn't scale well.
- Web.py: Too basic.

And the winner is…

- Team has experience using Tornado.
- Tornado suited for high throughput applications.
- Removes threading bottleneck.

Tornado Migration Progress

1. Converted main server loop, handlers to Tornado.
2. Converted index template to Tornado templating style.
3. Migrated RabbitMQ workers onto Tornado I/O loop.
4. New configuration parsing mechanism.

UI Enhancement

- Current web implementation at http://499.github.io/shoal/
- Implementation Progress
  - D3: Data-Driven Documents to display geologic location of available SQUID servers on a map
  - Using Websockets to make the map dynamic

QUESTIONS?