Ram Vakada

Prof. Alexey Nikolaev – Functional Programming in OCaml

Final Project Proposal

March 28th, 2019

Project Name: OCamlAlert

Proof Of Concept: https://github.com/RamVakad/OCamlAlert/tree/PoC

Design Changes: Project will no longer be using Lwt. It will just use plain Unix Sockets & Threading. Client module will ping the server module every 5 seconds for alerts until termination. Admin module is simplified to just accept two command line arguments and send a single message to the server.

This is the final project proposal. The proof of concept available at the link above currently has two modules: the simple server & simple client. The server accepts connections on port 8484 and sends out 10 messages to every client before closing the connection. The client in return opens a connection and endlessly accepts messages until it receives the message "[END]" from the server.

Revised Proposal

OCamlAlert is a TCP based alert system. The goal of OCamlAlert is to design an alert system like CUNY's alert system. CUNY's alert system works in such a way that all registered users are sent a text message informing them about emergencies like the school being closed down due to severe weather of if there is an active shooter and even fires. In the case of OCamlAlert, any user that is connected to the server will be able to receive an alert that an administrator sends out.

The OCamlAlert system will consist of three modules:

1. Server
2. User Client
3. Admin Client

The server module will no longer use Lwt for concurrency. The server on initialization will open a server socket on port 8484. Upon any connection request, the first communication between the server and the client will be a handshake. The handshake will be used to determine whether the client is a user or an admin. If it is a user, the client will send the current alert set on the server. If the client is an administrator, a special Admin Client handler will be assigned to the connection. If the admin client handler receives data from the open connection, the data will be set as the new alert which will then be sent out to all the connected clients.

The User Client module is a simple OCaml program that connects to the server on port 8484, identifies itself as a user through the initial handshake, and just waits for alerts. Any alert received will simply be printed on screen and the client will continue listening until termination.

The previously proposed Admin Client was going to accept user input for the admin password and alerts. The new Admin Client will simply accept two system arguments, the password and the alert to be set and just send that message to the server.