Christina Nguyen: cpn716@vt.edu

Nathaniel Hughes: njh2986@vt.edu

ECE 4564, Assignment 2

Section 1 – Objectives

Assignment 2 aimed to use Python network programming to have two Raspberry Pi’s communicate with each other via a shared message queue. The first Raspberry Pi (Pebble) was to act as a client that parses the command line in order to assemble / produce messages (pebbles) to the second raspberry pi (Bottle). The messages are consumed by the Bottle which processes the contents and responds the client. The server pi also controls an array of four LED’s to indicate the number of messages in the bottle. Here are our primary implementation objectives:

The first objective is to use the Advance Message Queuing Protocol (AMQP) implementation of RabbbitMQ to facilitate the queuing of messages from client to server. Another objective is to utilize a remote procedure call (RPC) for processing the client’s request / message. A third objective is to enforce data persistence: all of the messages sent from the client are shelved in a datastore on the server side, and all of the message responses received from the server / bottle are stored in a datastore on the client side. A fourth objective is to send all of the messages in JSON format, which requires conversion from dictionary to JSON object and vice versa. Another objective of this project is to display the number of messages in the bottle via LED’s connected at the server’s GPIO pins.

Section 2 – Team Member Responsibilities

Nathaniel primarily worked on the RPi 1 (Pebble), parsing the command line, generating messages, and sending those messages over to the server/bottle. His Pi made a remote procedure call to have the message processed on the server side. GetOpt is used to parse the command line arguments. The client pi then sends an assembled message to the server and awaits a response. Upon receiving a response, the response message is checked for validity and then stored in a persistent datastore.

Christina worked mainly on the server side, i.e. the Bottle. She wrote the remote procedure call and made it available to the client; the RPC processes the message received from the client and does one of three actions after parsing the message: the RPC could initiate a pull, a pullr or a push. A push means that a pebble is being added to the bottle, as well as persistent storage. A pull means that a pebble is being removed from the bottle and returned to the client, while a pullr is a pull, except the pebble is copied — not removed — from the bottle. Christina implemented each of these actions, as well as the querying functionality associated with the pull and pulls actions. Finally, Christina build the circuit for LED indication of pebble count in the bottle.

Section 3 – Conclusions

Overall, this project went smoothly, but there were a couple of obstacles faced: primarily we were challenged by setting up zeroconf, as well as getting initial communications set up via the rabbitMQ server and indicating the pebble count in the bottle via the LED’s.

The rabbitMQ server had to be installed and configured for use in this project. A new user was created and its permissions set. The server must be running in order for the server and client to utilize the message queue. One concern that we addressed was over whether or not messages were being properly consumed, allowing for the queue to be cleared after time.

zeroconf is a means of broadcasting device information on a local area network (LAN). It’s use for this assignment is to remove the need for barcoding the server’s IP address in the client-side code. We faced challenges installing an implementation of zeroconf and broadcasting the server’s information for the client to subscribe to…

This assignment was a good experience that offered a lot to learn in the area of network programming, specifically concurrency and socket communication in Python. Hardware malfunctions, requiring SD card reimaging, proved to be a headache. The team is still working on building a more permanent interface to each Raspberry Pi, by employing static IP’s. This assignment served as a good introduction to network applications, the Twitter API and the Raspberry Pi platform.