CPSC 4240: MS3

Alexander Sferrella, Steven Nix, Ronnie Funderburk

Project Summary:

This project uses node-js, Mysql, MQTT, D3, Express Web Server, and various dependencies to get the cpu load averages from the client computer, store them on the server in a mysql database, graph them with D3 and display them using an Express Web Server in the form of a web page. The client will scan a computer and report cpu load averages and timestamps. This data is then sent to a node.js server using MQTT. The server will then take that data and store it into a mysql database after which an Express Web Server will host a website that will act as a front end user interface displaying the generated graphs in the user's web browser.
This assignment is based on a similar project from another class, but the original project was much less in-depth.

Motivations:

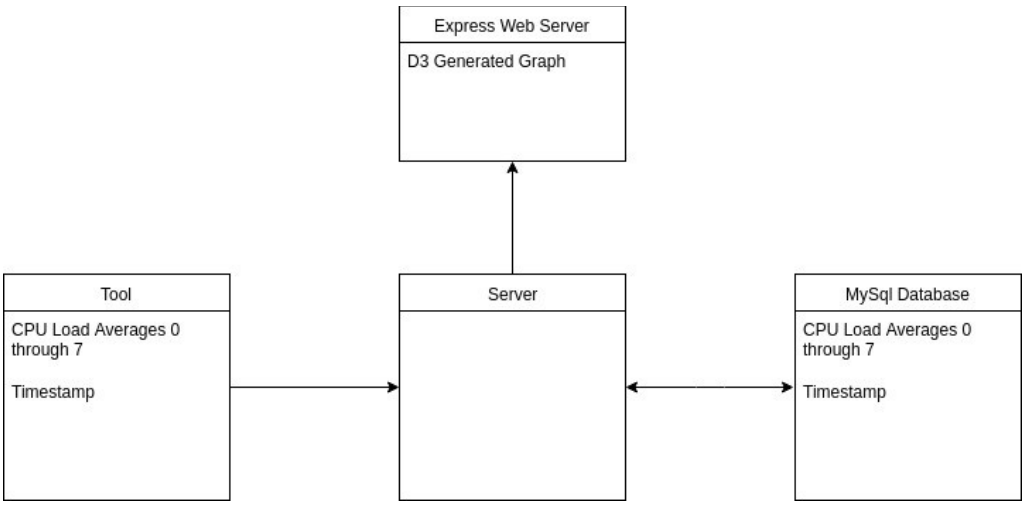
Our primary motivation for this project is to complete all requirements. In order to accomplish this, we originally decided to use an Raspberry Pi for our server, in order to get some experience with using the device. But due to time constraints and conflicts, we decided against this and decided to host our server on Ubuntu running a laptop.

Milestones:

- Milestone 1: Figure out what tools would be best used for this program.
  - MySql. Node-JS, MQTT, D3, Express Web Server.
- Milestone 2: Take original code and turn it into a client and make a server that can accept data from it.
- Milestone 3: Connect client and server with mqtt, and connect to Mysql Database.
- Milestone 4: Use D3 to make user interface.

High level design:

The tool sends data to the server using mqtt. The server then stores the data on a Mysql database. Due to lack of experience with php the server also generates a csv file that is used with the user interface web page and d3 to make 8 graphs 1 for each cpu.



Security:

Mysql: We hardened the mysql server using mysql_secure_installation. This includes setting a strong randomly generated password for root, removing anonymous users, Disallowing root login remotely, and removing test databases,

Client and Server: I use the crypto-js dependency and encrypt all data coming from the client using AES-256 and a randomly generated key and then decrypt it on the server side and store it in the MySql Database.

Server (ExpressJS Webserver): We Installed the helmet module which provides protection against well known web vulnerabilities by setting the HTTP headers appropriately. Helmet does this by installing nine smaller middleware functions that set security-related HTTP headers. This includes csp, hidePoweredBy, hpkp, hsts, ieNoOpen, noCache, noSniff, frameguard, xssFilter.

Dependencies: Dependencies give nodejs a lot of power however they can also be a source of insecurity so I installed two vulnerability scanners nsp check and snyk. Nsp check will only check for vulnerabilities and give suggestions. Snyk will scan for vulnerabilities and then patch or update for you.

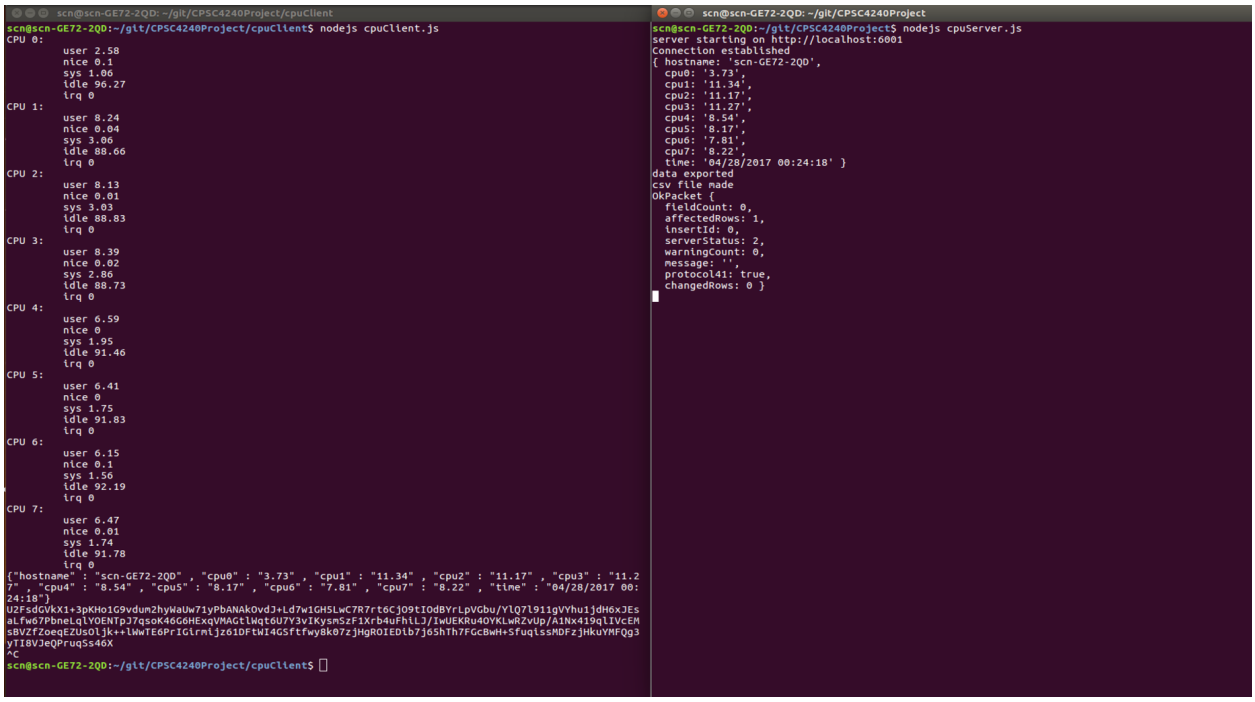
Validation:

For the cpu load averages the only way to validate right now is to compare the server console log and the sql database. Both the Server and the Client are validated and the MySQL database is storing and querying as it should. The user interface also shows 8 line graphs 1 for each cpu on an individual html page acting as an user interface.

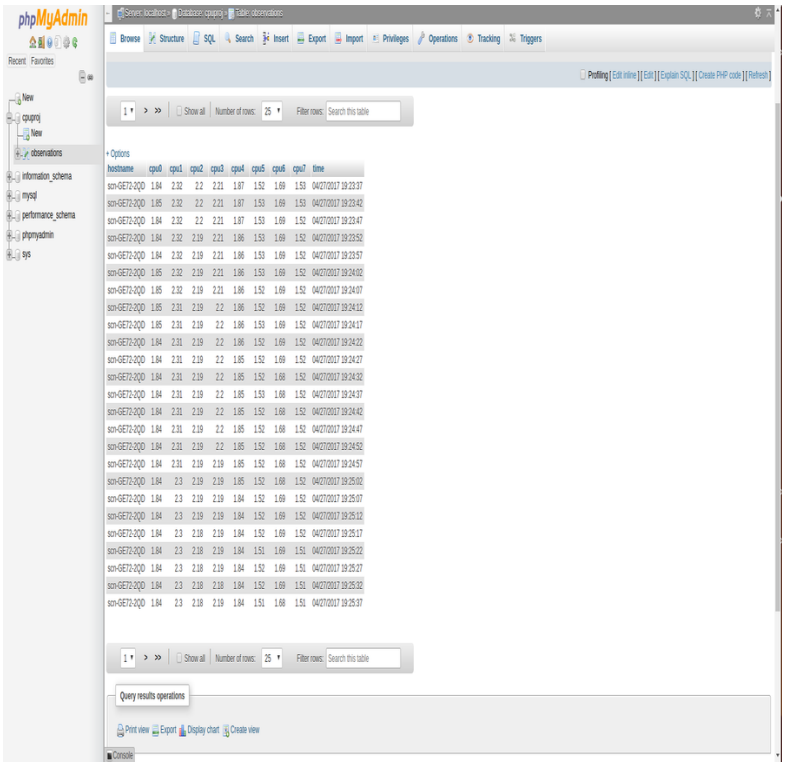
Submission details:

- cpuClient
  - cpuclient.js
  - package.json
- public
  - index.html
  - data
    - log.csv
  - other non important files for expressjs
- cpuServer
  - cpuserver.js
  - package.json

Client and Server Demo:



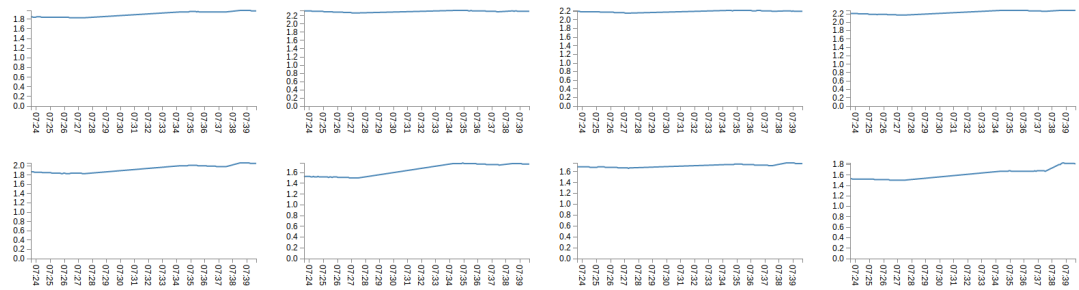
Mysql Demo:



User Interface Demo:

CPSC 4240 Group Project

By: Steven Nix, Ronnie Funderbook, Alex



X axis is the date/time of data point

Y axis is the cpu load percentage

The graphs go in order starting with cpu 0 - cpu 4 in the top row and cpu 5 - cpu 8 in the bottom

Due to securty issues please run in any browser except chrome

To run in chrome you must allow the d3 plugin to run (google for more info)

Contact information: scn@g.clemson.edu.