

Assignment 2

Report

ECE 4564 - Group 20

Dr. Plymale

February 27, 2017

Authors:

Hakeem Bisyr - hbisyr2@vt.edu

Jason Britton - bjason1@vt.edu

Andrew Bryant - andrewpb@vt.edu

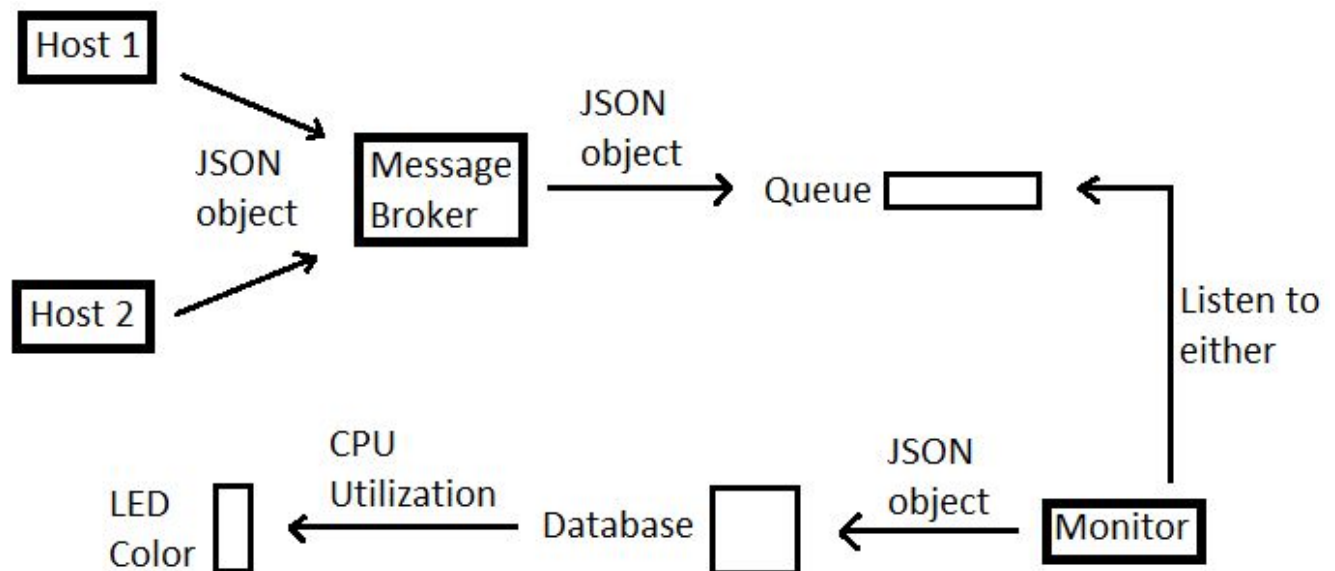
Josh Rehm - jrehm135@vt.edu

Section 1 – Objectives:

Two Host Raspberry Pis record performance data on the CPU utilization and the network throughput. This data is published in JSON notation using RabbitMQ to Message Exchanges on a Broker Raspberry Pi. The Host Pi's script must be given the Message Broker Pi's IP address and a routing key, which is given using command line arguments. The Host Pi's script also can take in specific virtual host and user login credentials, but if not specified they default to the root virtual host and guest login.

The Message Broker Pi manages message queues, and stores and routes the messages using RabbitMQ. The Message Broker subscribes to the two Hosts' exchanges, and publishes the data to the Monitor Pi. The Monitor Pi subscribes to the metric data on the Monitor Pi, and remotely monitors the Hosts' performance. The performance data is stored in Mongo database. Current CPU threshold alerts are displayed on an RGB LED using the Monitor's GPIO pins.

System Diagram



Section 2 – Team Member Responsibilities

Jason Britton - I created software to run on the monitor device. The program connects to the RabbitMQ server using the parameters given through command line. It subscribes to the specified queue and for each JSON message it receives, it stores the data in a MongoDB database, as well as comparing the data to the highest and lowest values that have been received. The database is then updated with the highest and lowest values for each category, and all of the data is printed to the standard output. Finally, the GPIO code for the hardware is run.

Andrew Bryant - I worked on the LED code and hardware for the monitor Raspberry Pi. I also worked on the RabbitMQ publishing code and creating the JSON objects for the Host Raspberry Pis, as well as some error handling.

Josh Rehm - My part involved getting the server up and running properly, as well as monitoring the exchanges between the host raspberry pis and the monitor raspberry pi. As a generalization, this role involved extensive testing of the pis to confirm proper operation and transfer, as well as figuring out guest access from a remote host.

Hakeem Bisyrir - I was put in charge of calculating the system timing information and calculating the utilization and network throughput. By accessing the correct files and gathering relevant information, I was able to access the system's uptime, idle time, and bytes sent and received per second. I was also in charge of gathering the command line information to connect to the message broker and error handling for the host pis.

Section 3 – Conclusions:

After many hours, we successfully completed the project. The program works as designed and successfully sends and receives messages from the queue. The messages sent from the host pis send their JSON messages to a message broker pi that stores the information. A monitor pi then binds the information into a queue and subscribes to them. One of the hardest challenges was successfully calculating the utilization from the correct file. After reading from the uptime file in the proc directory, the utilizations would occasionally turn out negative. After realising this, we read the files from the stat file which correctly calculated the correct number between 0 and 1. Overall this was a successful assignment and our teamwork and communication skills allowed us to complete it with correct specifications.