CS352 Project 2: Load-Balancing across DNS servers

Karun Kanda (kk951), Shila Basu (sb1825)

April 2th, 2021

1 Questions

1.1 Please write down the full names and netids of both your team members.

Team Member 1: Karun Kanda - kk951
Team Member 2: Shila Basu - sb1825

1.2 Briefly discuss how you implemented the LS functionality of tracking which TS responded to the query and timing out if neither TS responded.

We implemented the LS by keeping two dictionary objects to keep track of the routed messages: one to store message queues and the other to keep track of requests. The message queue collection used socket objects as keys and stored all inbound data into a message queue class for each readable socket. It then, dequeued messages for each writable socket to send to its final destination. Upon a client request, the LS server would store the start time and hostname into a data collection referenced by the inbound message. The LS server would then start polling this collection which would bring about one of the following outcomes:

1. If TS1 of TS2 sends a reply, LS would update the client request data structure with the reply message and source. On the next polling iteration, it would then read the reply from this data structure and relay it to the client.
2. If the client request structure has no reply and the current time is greater than five seconds from the start time, LS sends back ”Hostname - Error: HOST NOT FOUND” to the client.
3. If the client request structure has no reply and the current time is less than five seconds from the start time, LS will add the message back to the queue and wait for the next iteration to poll again.

1.3 Are there known issues or functions that aren't working currently in your attached code? If so, explain.

All the functions in the program should be working according to the given specifications.

1.4 What problems did you face developing code for this project?

The main problems came when debugging the code and dealing with blocking sockets. For each issue encountered, all four processes had to be restarted in order and the TS servers required restarts for each terminal session. There was also an issue when LS read the client socket in a delayed manner. All the messages piled up in the socket, leading LS to read one large message with the hostnames concatenated together.

1.5 What did you learn by working on this project?

We learned that using batch scripts were useful in saving time while debugging the code. Also, implementing a number of print statements to benchmark the execution flow allowed for us to catch errors and debug the code in real-time. Lastly, we learned how the select.select() function worked. After calling it with a list of input and output sockets, it returned a list of all readable and writeable sockets once it was triggered by an event on one of the sockets.