Darren Eam
SID: 1536854

Design Document for Assignment 3:

Idea: make a program that listens to incoming connections on one port, and disperses those connections to other servers on specified ports.

Command: “loadbalancer [-N ___] [-R ___] (listening_port) (server_port1 [server_port2 …])

- -N: number of parallel connections to maintain
- -R: number of requests to process between each mandatory health check
- Listening_port: the port number we receive connections from
  - MUST BE THE FIRST ARG (not including flags)
- Server_ports: the port numbers we send the connections from the listening port to
  - Any argument that isn’t the first

Restrictions/Details:

- Port number for listening must be included in command line
- At least one port for servers must be included in command line
- If a server doesn’t ***begin responding*** to a request in ***5 seconds***, then send a ***500 error*** to the client that’s waiting on that server

Globals: Job queue, port array 32x3, [portnum, errors, entries], mutex/semaphores, counter variables

Subroutines/Modules:

- int* health_probe(int serverports[m][n], int index)
  - Send healthcheck to the server
  - Parse the response
  - Pass back the errors and entries values
- Int find_server()
  - Search thru global ports array
    - If server has 0 requests, use immediately
    - If it has a minimum num of requests seen, keep track of that
    - If == the current minimum num of requests, use the one w the least errs
- Void worker_thread()
  - Sem wait, lock
  - Dequeue and find the best server
  - Unlock, sem post
  - Send to the best server

Darren Eam
SID: 1536854

Main():

- Getopt() to check command line for necessary arguments
    - -N: number of connections we can send to diff servers in parallel
    - -R: number of requests b/w health checks for each server
    - Else
        - If this arg is negative: exit in error
        - If this arg is the first, set it as the load balancer port number
        - Else this argument is a server port number to send
- First argument is the loadbalancer port; server_listen() on this
- For loop to get the rest of the arguments
    - Add this port to an array of ports (with healthcheck infos)
- while(1):
    - If we need to make a healthcheck to all servers (after -R requests to the LB)
        - (For loop to make one to all servers)
            - Get healthcheck info (health_probe())
            - Update values of port array
    - Accept a job
    - Sem wait, lock
    - Add job to queue
    - Unlock, sem post