FastChat

Thrice as Nice

Nov 25, 2022

CONTENTS:

CHAPTER

ONE

FASTCHAT

1.1 client module

client.**main**()

This is the main function Firstly, a socket connection is initiated between loadbalancer and client, when the user logs in, a server is alloted to it. Client disconnects its connection with the loadbalancer and a socket connection is initiated between the client and the server alloted to it All the messages received by the client while it was offline are displayed and receiving and sending threads are initiated.

client.**receiving_func**()

It is thread for receiving information from the server Depending on the code received from the server, the parameters will be changed accordingly. The codes received from the server have the following meaning:

1. c: Receive another code from the server
2. y: Message has been sent successfully
3. n: Unable to send direct message
4. l: Group is full and hence no further member can be added
5. t: User exists already in the group
6. e: Receiving public key of receiver
7. k: Receiving private key of group
8. p: Decrypting group private key using private key of user
9. s: Server is going to send the usernames of all the members of the group to the admin
10. u: Have to encrypt the message of sender using the public key of receiver sent by server and have to send it back to the server
11. g: Have to encrypt the message of sender using the public key of group sent by server and send the encrypted message back to server
12. a: Received an image file
13. b: Sent an image file
14. q: Client has logged out

client.**user_interface**(*display_menu=0*)

Function to display the user interface It is the thread for sending information to the server Different user intefaces are displayed as per the value of the display_menu parameter The parameter display_menu can take the following values and the following options are displayed as per the value of the display_menu:

1. display_menu = 0 : Main Menu

    1. g: for managing groups
    2. b: to send group message
    3. d: to send direct message

4. l: to logout

2. display_menu = 1 : Group Settings

    1. n: to create a new group
    2. m: to manage an existing group
    3. q: to go to previous menu

3. display_menu = 2 : Manage Existing Group

    1. a: to add a new member
    2. r: to remove a member
    3. s: to see all members in the group
    4. q: to go to previous menu

4. display_menu = 3 : Group message

    1. t: to type a message
    2. i: to send an image or text file
    3: q: to go to previous menu

5. display_menu = 4 : Direct message

    1. t: to type a message
    2. i: to send an image or text file
    3. q: to go to previous menu

Parameters **display_menu** (*int*) – contains information about which display menu is to be shown

1.2 loadbalancer module

loadbalancer.**clientthread**(*conn*, *addr*)

This is the receiving thraed of the loadbalancer which receives code and performs the respective function The loadbalancer receives the following codes which have the following meanings:

1. s: The following can be received after it which have the following meanings

    1. ci: Send the public key of the user
    2. cg: Send the public key of the group
    3. cs: Send the ip address and port number of the server to which the user is connected to
    4. cl: The user has logged out and thus should be deleted from the loadbalancer dictionary
    5. ag: Store the public key of the group in a dictionary of the load balancer

2. c: Perform sign-up and login of the user and stores its encrypted private key and public key in a dictionary

Parameters

- **conn** (*socket.socket*) – It is the socket object of the client/server which wants to establish socket connection to load balancer
- **addr** (*tuple*) – It is the address tuple of the client/server (ip,port) which wants to establish connection with the load balancer

loadbalancer.**least_connection**(*server_list*)

Returns the port number and ip address of the server with the mminimum number of clients

Parameters **server_list** (*list[int]*) – List of all servers

Returns returns the port number and ip address of the server with minium number of clients

Return type tuple

loadbalancer.**main**()

Accepts a connection request and stores two parameters,conn which is a socket object for that user, and addr which contains the IP address of the client that just connected Additionally, the databases for storing the credentials of all the clients, information of groups, individual messages for offline users and encrypted private keys of clients and their public keys are stored

loadbalancer.**round_robin**(*iter*)

Implements round robin algorithm to return the pointer to the next server to which the incoming client would get connected to

Parameters **iter** (*itertools.cycle*) – it points to the server to which the next incoing client will be connected to

loadbalancer.**select_server**(*server_list*, *algorithm*)

This function returns the tuple object of the server to which the incoming client would get connected to

Parameters

- **server_list** (*list of the tuples containing port and ip of each server*) –
- **algorithm** (*str*) – This is the algorithm which needs to be followed to select the server

Returns Returns the tuple object (ip,address) of the server to which the incoming client will be connected

Return type tuple containig ip and address of the server

1.3 server module

server.**clientthread**(*conn*, *addr*)

This is the thread which receives messages from the client and responds accordingly. Messages are sent from the server to the clients connected to that particular server. The following codes are received by the server which have the following meaning:

1. cg: Check if the group already exists or not
2. ci: Check if the individual exists or not
3. ng: Create a new group with the client sending the code as admin
4. eg: Check if the group exists and the client sending the code to the server is the admin

5. ai: Add an individual to the group
6. ri: Remove an individual from the group
7. sa: Send the names of all the members of the group to the admin of the group
8. wg: Send the text message in encrypted format to an individual
9. ig: Send an image in an encrypted format to a group
10. wi: Send the text message in encrypted format to an individual
11. ii: Send an image in encrypted format to an individual

Parameters

- **conn** (*socket.socket*) – It is the connection object of the client which has established a socket connection with the server
- **addr** (*tuple*) – It is the address (ip,port) of the client which has established a socket connection with the server

server.**letsconnect**(*ip*, *port*)

This function establishes socket connection between the server to the server with the ip and port passed as parameter The following codes are received by the server which have the following meanings:

1. wi: Send the text message in encrypted format to the server to which the receiver client is connected
2. ii: Send the image in encrypted format to the server to which the receiver client is connected
3. wg: Send the text message sent by a client in a group in encrypted format to the server to which the receiver client (group member) is connected
4. ig: Send the image sent by a client in a group in encrypted format to the server to which the receiver client (group member) is connected
5. gk: Get the encryoted group private key from the loadbalancer

Parameters

- **ip** (*str*) – It is the ip address of the server which wants to establish a socket connection with this server
- **port** (*str*) – It is the port number of the server which wants to establish a socket connection with this server

server.**main**()

Accepts a connection request and stores two parameters, conn which is a socket object for that user, and addr which contains the IP address of the client that just connected It also initiates the client thread

server.**remove**(*connection*, *usr*)

Removes the connection of the user with the server and loadbalancer that is change the status of the user from online to offline The entry in the dictionary with the key as the username of the user is deleted from both the server to which the client is connected and the loadbalancer

Parameters

- **connection** (*socket.socket*) – Connection with the user which has logged out which needs to be removed
- **usr** (*str*) – The username of the user who has logged out

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX